Received: 29 December 2018 | Revised: 17 March 2019 | Accepted: 3 May 2019 | Published on: 4 June 2019

DOI: 10.1002/nav.21850

RESEARCH ARTICLE

WILEY

On gamma estimation via matrix kriging

Xin Yun[1] | **L. Jeff Hong**[1,2] | **Guangxin Jiang**[3] | **Shouyang Wang**[4]

[1]School of Data Science, Fudan University, Shanghai, China
[2]School of Management, Fudan University, Shanghai, China
[3]School of Management, Shanghai University, Shanghai, China
[4]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

Correspondence
Guangxin Jiang, School of Management, Shanghai University, Shanghai, China.
Email: gxjiang@shu.edu.cn

Funding information
National Natural Science Foundation of China, 71801148. Research Grants Council, University Grants Committee, 11504017, 16203214. Shanghai Municipal Education Commission, N.60-D129-18-202.

Abstract

In financial engineering, sensitivities of derivative prices (also known as the Greeks) are important quantities in risk management, and stochastic gradient estimation methods are used to estimate them given the market parameters. In practice, the surface (function) of the Greeks with respect to the underlying parameters is much more desired, because it can be used in real-time risk management. In this paper, we consider derivatives with multiple underlying assets, and propose three stochastic kriging-based methods, the element-by-element, the importance mapping, and the Cholesky decomposition, to fit the surface of the gamma matrix that can fulfill the time constraint and the precision requirement in real-time risk management. Numerical experiments are provided to illustrate the effectiveness of the proposed methods.

KEYWORDS

financial risk management, gradient estimation, Greeks, stochastic kriging

1 | INTRODUCTION

In financial engineering, sensitivities of derivative prices, also known as the *Greeks*, are important quantities in risk management. Based on the sensitivity of the derivative price with respect to a small movement of a given underlying market parameter, we can isolate the risk, and trade the related underlying assets or instruments to achieve a desired risk exposure (Björk, 2009). Hedging strategies are often created to manage the risk of a derivative portfolio based on Greeks. For instance, delta hedging and delta-gamma hedging are two commonly used hedging strategies that are designed to reduce the risk from the fluctuation of the underlying assets, where delta and gamma are the first-order and second-order sensitivities[1] of the derivative price with respect to the price of the underlying asset, respectively. These strategies make delta and gamma particularly important among the Greeks.

If the analytical formula of the price of a derivative is known, we can easily obtain the Greeks by just taking partial derivative with respect to the corresponding underlying parameter. When the analytical formula is unavailable, stochastic gradient estimation methods are powerful alternatives for sensitivity analysis (see Fu, 2006) for a thorough introduction). Finite-difference approximation provides a "brute force" approach to gradient estimation and is easy to implement (Glasserman, 2013). It is commonly used when we only have simulation outputs. If we know some information about the payoff function or the probability density of underlying assets, pathwise method and the likelihood ratio method are two commonly used approaches. The pathwise method (or perturbation analysis) is originated by Ho and Cao (1983), and later elaborated by Broadie and Glasserman (1996) and Fu and Hu (1997) in financial applications. It typically requires the payoff function to be continuous. The likelihood ratio (or score function) method is proposed by Glynn (1990) and Reiman and Weiss (1989). It requires no continuity of the payoff function by differentiating the probability density function instead of the payoff function. However, its estimators typically have larger variances than those of the pathwise method. As improvements and complements of the pathwise and the likelihood ratio method, some other methods, for example, kernel method (Elie, Fermanian, & Touzi, 2007; Liu & Hong, 2011), weak derivatives method (Heidergott, Vazquez-Abad, Pflug, & Farenhorst-Yuan, 2010;

[1]Here, we use first-order and second-order sensitivities instead of first-order and second-order derivatives to distinguish them from the derivatives as financial products.

Pflug, 1989), and generalized likelihood ratio method (Peng, Fu, Hu, & Heidergott, 2018; Wang, Fu, & Marcus, 2012) are proposed.

In practice, the Greeks need to be estimated in real time, so that the risk managers can hedge the risk of their portfolios when market parameters change. For instance, when using the delta-gamma hedging strategy, the prices of the underlying assets (e.g., stock prices) may change frequently and one has to adjust the portfolio in real time to hedge the risk. If the analytical formulae of the prices of the derivatives included in the portfolio are available, one can obtain the delta and gamma immediately by taking partial derivatives directly. For example, for a European call option, the Black-Scholes formula provides the analytical formula of the price, coupled with the Greeks. It is an important reason why the Black-Scholes model is popular among financial practitioners, even though many point out that the Black-Scholes model is oversimplified (Schoutens, 2003). For portfolios including exotic derivatives with complicated underlying asset models, price formulae are typically unavailable, and therefore calculating the delta and gamma in real time becomes a challenging task. The stochastic gradient estimation methods mentioned above cannot be applied directly due to the large computational burden, even though they can achieve any degree of precision by simply increasing the simulation sample size. The computation time becomes an even more significant issue if the portfolio includes derivatives with multiple underlying assets, for example, basket option, rainbow option, and quanto option. In this setting, the delta is a vector and the gamma is a matrix, which apparently needs more computation time than a scalar.

In this paper, we propose a series of approaches to obtain the gamma matrix surface, under the offline-simulation-online-application (OSOA) framework proposed by Hong and Jiang (2018) (also see Jiang, Hong, & Nelson, 2019). The matrix surface here means that the generalized surface that each "point" on the surface is a matrix. Under the OSOA framework, we treat simulation as a data generator, apply state-of-the-art analytic tools to build predictive models, and then use the predictive models for real-time applications. The OSOA framework can be divided into two stages. In the offline-simulation stage, we conduct a large amount of simulation experiments on different scenarios of the parameters, then we use state-of-the-art data analytics tools to build a predictive model. In the online-application stage, the predictive model acts like a simple model with closed-form expressions and is used in real-time problems once the parameters are observed. The OSOA framework is particularly suitable for financial applications, since there is a large amount of time for offline simulation when the financial market closes, so one can choose the scenarios based on the present market conditions to build a predictive model for the next trading day. In this paper, we focus on the gamma matrix surface, and use it in real-time risk management problems. In the offline-learning stage, we select some scenarios of the underlying asset prices, and estimate the corresponding gamma matrix for each scenario via stochastic gradient estimation methods. We then use interpolation methods to build the gamma matrix surface. Specifically, we use the stochastic kriging method, see Ankenman, Nelson, and Staum (2010), Chen, Ankenman, and Nelson (2013), Kleijnen (2009), and van Beers and Kleijnen (2003). However, the stochastic kriging methods in the literature are designed only to handle scalars. In this paper, we consider how to extend it to handle matrices. Moreover, the gamma matrix surface may have special structures, which make the fitting problem more challenging. For example, if the price formula is convex with respect to the prices of the underlying assets, the gamma matrix is positive definite for any fixed prices of the underlying assets. We also propose methods that can maintain the positive definiteness. Notice that the proposed methods can be applied not only in estimating Greeks, but also in other real-time sensitivity analysis or stochastic gradient estimation problems.

The rest of this paper is organized as follows. We first review the pathwise Greeks estimation and stochastic kriging method in Section 2. Three matrix kriging methods, element-by-element (EE), importance mapping, and Cholesky decomposition methods, are proposed to estimate the gamma matrix surface in Section 3. Numerical results are presented in Section 4, followed by conclusions in Section 5.

2 | PRELIMINARIES

In this section, we review the ideas of the pathwise sensitivity estimation and stochastic kriging in Sections 2.1 and 2.2, respectively. These ideas are used in later section in building gamma matrix surfaces.

2.1 | Pathwise method

The pathwise method was originally proposed to estimate sensitivities of queueing systems, and then extended to estimating Greeks; see Broadie and Glasserman (1996). In this subsection, we recall the basic results of infinitesimal perturbation analysis (IPA) and smoothed perturbation analysis (SPA), as the foundations of the following proposed methods.

Suppose that the discounted payoff function of a derivative $Y(\boldsymbol{\theta})$ is a random variable defined on probability space $(\Omega, \mathcal{F}, P)$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_d)^\top \in \Theta \subseteq \mathbb{R}^d$ is the underlying parameters, and we need to take derivative with respect to them. The price of the derivative equals to $\mathbb{E}[Y(\boldsymbol{\theta})]$, and the sth-order Greeks can be defined by $\partial^s \mathbb{E}[Y(\boldsymbol{\theta})]/\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ldots\partial\theta_d^{l_d}$, where $0 \le l_i \le d$, $i = 1, 2, \ldots, d$, and $\sum_{i=1}^d l_i = s$, $s = 1, 2, \ldots$. If we can interchange the differentiation and the expectation, that is

$$\frac{\partial^s \mathbb{E}[Y(\boldsymbol{\theta})]}{\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ldots\partial\theta_d^{l_d}} = \mathbb{E}\left[\frac{\partial^s Y(\boldsymbol{\theta})}{\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ldots\partial\theta_d^{l_d}}\right],$$

then the unbiased IPA estimator of the corresponding Greek is given by $\partial^s Y(\boldsymbol{\theta})/\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ \ldots\ \partial\theta_d^{l_d}$.

The interchange of the differentiation and the expectation is valid if the discounted payoff function is Lipchitz continuous. For instance, the discounted payoff function of a European call option $Y(S(0)) = e^{-rT}\{\max S(T) - K, 0\}$ is Lipchitz continuous with respect to $S(0)$, where $S(0)$ is the initial value of the underlying asset price, $S(T) = S(0)\exp((r - 1/2\sigma^2)T + \sigma W(T))$ is the underlying asset price at maturity T, $W(T)$ is a Brownian motion observed at time T, K is the strike price, σ is the volatility of underlying price, and r is the risk-free interest rate. In this case, $Y(S(0))$ is differentiable with respect to $S(0)$ with probability 1, so the unbiased IPA estimator of delta is given by $\partial Y(S(0))/\partial S(0) = e^{-rT}S(T)\mathbf{1}_{\{S(T)>K\}}/S(0)$. However, the IPA estimator of gamma cannot be obtained due to the discontinuity of the indicator function in $\partial Y(S(0))/\partial S(0)$ with respect to $S(0)$.

To overcome the discontinuity, we can apply SPA that smooths $Y(\boldsymbol{\theta})$ via conditioning. Basically, SPA first smooths the discounted payoff function by taking conditional expectation on some proper random variables or events Z, that is

$$\frac{\partial^s \mathbb{E}[Y(\boldsymbol{\theta})]}{\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ \ldots\ \partial\theta_d^{l_d}} = \frac{\partial^s \mathbb{E}[\mathbb{E}[Y(\boldsymbol{\theta})|Z]]}{\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ \ldots\ \partial\theta_d^{l_d}}.$$

Notice that $Y(\boldsymbol{\theta})$ itself may be discontinuous, but the conditional expectation $\mathbb{E}[Y(\boldsymbol{\theta})|Z]$ may be Lipchitz continuous. Then we can interchange the differentiation and the expectation to obtain the unbiased SPA estimator $\partial^s \mathbb{E}[Y(\boldsymbol{\theta})|Z]/\partial\theta_1^{l_1}\partial\theta_2^{l_2}\ \ldots\ \partial\theta_d^{l_d}$. Notice that the key to SPA is to find an appropriate random variable Z to condition on.

Remark 1 In this paper we use SPA to estimate the gamma, which performs quite well and can reach any degree of precision by increasing the sample size. However, we can also use the likelihood ratio method, or other gradient estimation methods to estimate the gamma. After estimating the gammas at different scenarios of the underlying asset prices, we use the methods proposed in Section 3 to build the gamma surface.

2.2 | Stochastic kriging metamodel

Ankenman et al. (2010) proposed a stochastic kriging approach to construct the surface of scalar simulation outputs. Suppose that we are interested in modeling an unknown surface $y(\mathbf{x}) \in \mathbb{R}$, and $\mathbf{x} = (x_1, x_2, \cdots, x_d)^\top \in \mathbb{R}^d$ denotes a design point (think a scenario of the underlying asset prices). In the stochastic simulation, the hth simulation replication at the design point $\mathbf{x}$ is modeled as

$$\mathcal{Y}^h(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top\boldsymbol{\beta} + \mathsf{M}(\mathbf{x}) + \varepsilon^h(\mathbf{x}), \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is a $p\times 1$ vector of preset functions of $\mathbf{x}$, representing the prior information of the unknown surface, and $\boldsymbol{\beta}$ is a $p\times 1$ vector of unknown parameters. M is a realization of second-order stationary Gaussian random field with mean 0. Usually, if we do not know any information about $\mathbf{f}(\mathbf{x})$, we can simply let $\mathbf{f}(\mathbf{x})^\top\boldsymbol{\beta} = \beta_0$. The noise terms $\varepsilon^1(\mathbf{x})$, $\varepsilon^2(\mathbf{x})$, … are independent and identically distributed normal distribution $\mathrm{N}(0, V(\mathbf{x}))$ at each design point $\mathbf{x}$, independent of M, and the variance $V(\mathbf{x})$ only depends on the design point $\mathbf{x}$.

Furthermore, we choose N design points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ in the design space. Suppose that we replicate the experiment n_k times for the design point $\mathbf{x}_k$, $k = 1, 2, \ldots, N$. The sample mean of the simulation outputs and the simulation errors at the design point $\mathbf{x}_k$ are $\overline{\mathcal{Y}}(\mathbf{x}_k) = 1/n_k\sum_{h=1}^{n_k}\mathcal{Y}^h(\mathbf{x}_k)$ and $\overline{\varepsilon}(\mathbf{x}_k) = 1/n_k\sum_{h=1}^{n_k}\varepsilon^h(\mathbf{x}_k)$, $k = 1, \ldots, N$. Then the averaged response is given by

$$\overline{\mathcal{Y}}(\mathbf{x}) = \beta_0 + \mathsf{M}(\mathbf{x}) + \overline{\varepsilon}(\mathbf{x}).$$

Let $\mathbf{x}_0$ be a new point that we are interested in its response value $\mathsf{Y}(\mathbf{x}_0) = \beta_0 + \mathsf{M}(\mathbf{x}_0)$. Here we consider a linear predictor of $\mathsf{Y}(\mathbf{x}_0)$. Let Σ_{M} be the $N\times N$ variance-covariance matrix among the response values at N design points, that is, the (i, j)th element in the matrix $(\boldsymbol{\Sigma}_{\mathsf{M}})_{i,j} = \mathrm{Cov}(\mathsf{M}(\mathbf{x}_i), \mathsf{M}(\mathbf{x}_j))$. One can further assume that $\mathrm{Cov}(\mathsf{M}(\mathbf{x}_i), \mathsf{M}(\mathbf{x}_j)) = \tau^2 R(\mathbf{x}_i, \mathbf{x}_j;\boldsymbol{\theta})$, where $R(\mathbf{x}_i, \mathbf{x}_j;\boldsymbol{\theta})$ is the correlation that depends only on $\mathbf{x}_i - \mathbf{x}_j$, $i, j = 1, 2, \ldots, N$, and unknown parameter $\boldsymbol{\theta}$ according to the second-order stationary condition of M. Let $\Sigma_{\mathsf{M}}(\mathbf{x}_0, \cdot)$ denote the covariance vector between $\mathsf{M}(\mathbf{x}_0)$ and $\mathsf{M}(\mathbf{x}_i)$, $i = 1, 2, \ldots, N$, that is, the ith element in the vector $(\boldsymbol{\Sigma}_{\mathsf{M}}(\mathbf{x}_0, \cdot))_i = \mathrm{Cov}(\mathsf{M}(\mathbf{x}_0), \mathsf{M}(\mathbf{x}_i))$. Let $\boldsymbol{\Sigma}_\varepsilon = \mathrm{diag}[\mathrm{Var}(\overline{\varepsilon}(\mathbf{x}_1)), \mathrm{Var}(\overline{\varepsilon}(\mathbf{x}_2)), \ldots, \mathrm{Var}(\overline{\varepsilon}(\mathbf{x}_N))]$ be the covariance matrix of simulation noise terms. When the spatial parameters $\boldsymbol{\theta}$, τ^2, the noise term matrix $\boldsymbol{\Sigma}_\varepsilon$, and β_0 are known, the MSE-optimal predictor, which minimizes the mean squared error among all the linear predictors of the response $\omega_0(\mathbf{x}_0) + \boldsymbol{\omega}(\mathbf{x}_0)^\top\overline{\mathcal{Y}}$ at $\mathbf{x}_0$, is

$$\widehat{\mathsf{Y}}(\mathbf{x}_0) = \beta_0 + \boldsymbol{\Sigma}_{\mathsf{M}}(\mathbf{x}_0, \cdot)^\top[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\varepsilon]^{-1}(\overline{\mathcal{Y}} - \beta_0\mathbf{1}_N),$$

where $\mathbf{1}_N$ is the $N\times 1$ vector of ones and $\overline{\mathcal{Y}} = (\overline{\mathcal{Y}}(\mathbf{x}_1), \overline{\mathcal{Y}}(\mathbf{x}_2), \ldots, \overline{\mathcal{Y}}(\mathbf{x}_N))^\top$ is the $N\times 1$ vector of response sample means.

In general, $\boldsymbol{\theta}$, τ^2, β_0, and $\boldsymbol{\Sigma}_\varepsilon$ are unknown. Then, we can first estimate $\boldsymbol{\Sigma}_\varepsilon$ by

$$\widehat{\boldsymbol{\Sigma}}_\varepsilon = \mathrm{diag}\{\widehat{V}(\mathbf{x}_1)/n_1, \widehat{V}(\mathbf{x}_2)/n_2, \cdots, \widehat{V}(\mathbf{x}_N)/n_N\},$$

where

$$\widehat{V}(\mathbf{x}_k) = \sum_{h=1}^{n_k}(\mathcal{Y}^h(\mathbf{x}_k) - \overline{\mathcal{Y}}(\mathbf{x}_k))^2/(n_k - 1) \tag{2}$$

is the sample variance of response replications at design point $\mathbf{x}_k$, $k = 1, 2, \ldots, N$. Then $\boldsymbol{\theta}$, τ^2, and β_0 can be estimated via a maximum-likelihood estimation (MLE), and denoted by $\widehat{\beta}_0, \widehat{\tau}^2, \widehat{\boldsymbol{\theta}}$; see Ankenman et al. (2010). Therefore, the stochastic kriging predictor is given by,

$$\widehat{\mathsf{Y}}(\mathbf{x}_0) = \widehat{\beta}_0 + \widehat{\boldsymbol{\Sigma}}_{\mathsf{M}}(\mathbf{x}_0, \cdot)^\top[\widehat{\boldsymbol{\Sigma}}_{\mathsf{M}} + \widehat{\boldsymbol{\Sigma}}_\varepsilon]^{-1}(\overline{\mathcal{Y}} - \widehat{\beta}_0\mathbf{1}_N),$$

where $\widehat{\boldsymbol{\Sigma}}_{\mathsf{M}}(\mathbf{x}_0, \cdot)$ and $\widehat{\boldsymbol{\Sigma}}_{\mathsf{M}}$ are obtained by substituting $\widehat{\boldsymbol{\theta}}$ and $\widehat{\tau}^2$.

The stochastic kriging metamodel combines the intrinsic uncertainty that linear regression has assumed and the extrinsic uncertainty from simple kriging, and provides a powerful and flexible tool to model the whole response surface given limited information of design points and responses. However, it only applies to scalar response. In this paper, we consider the problem where the response is a matrix, and possibly a positive definite matrix.

3 | MATRIX KRIGING METHOD

Let $\mathbf{x} = (x_1, x_2, \dots, x_d)^\top \in \mathcal{X} \subseteq \mathbb{R}^d$ be a scenario of underlying asset prices (i.e., design point), where x_i is the price of the ith underlying asset, $i = 1, 2, \dots, d$, and $\mathcal{X}$ is the scenario space. Let $\mathcal{S}^d$ be the space of $d \times d$ symmetric matrix. Let

$$\mathbf{\Phi} : \mathbb{R}^d \to \mathcal{S}^d$$

be a mapping. The matrix surface can be defined by the mapping $\mathbf{\Phi}$, and the matrix surface fitting problem tries to measure the relationship between the scenario space, which is usually a vector space, and the corresponding response space, which is a matrix space. In this section, we focus on the gamma, which is a commonly used second-order Greeks in risk management, and propose a series of methods to build the gamma matrix surface via the stochastic kriging.

Suppose that the design points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ are given. For any $\mathbf{x}_k$, $k \in \{1, 2, \dots, N\}$, the corresponding gamma $\mathbf{\Phi}(\mathbf{x}_k)$ is estimated by a stochastic gradient estimation method, specifically, the SPA method. Let the hth simulation replication at the design point $\mathbf{x}_k$ be $\mathbf{H}^h(\mathbf{x}_k)$, which is modeled as $\mathbf{H}^h(\mathbf{x}_k) = \mathbf{\Phi}(\mathbf{x}_k) + \boldsymbol{\varepsilon}^h(\mathbf{x}_k)$, with $\boldsymbol{\varepsilon}^h(\mathbf{x}_k)$ being the simulation noise matrix. The noise can be related to the randomness of the payoff function and the gradient estimation method that we use. The (i, j)th elements in the simulation noise matrices $\varepsilon^1_{i,j}(\mathbf{x}_k), \varepsilon^2_{i,j}(\mathbf{x}_k), \dots$ represent the independent and identically distributed mean-zero sampling noise observed for each replication taken at design point $\mathbf{x}_k$, so $\mathbf{H}^h(\mathbf{x}_k)$ is an unbiased estimator of $\mathbf{\Phi}(\mathbf{x}_k)$.

Now we are interested in fitting the overall surface of the gamma, which is used to predict $\mathbf{\Phi}(\mathbf{x}_0) \in \mathcal{S}^d$ at any design point $\mathbf{x}_0 \in \mathcal{X}$ in real time. If $d = 1$, the problem degenerates to the classical setting, where the response is a scalar, then we can apply classical stochastic kriging in Section 2.2 to fit the surface. The more challenging setting is $d > 1$ (especially $d \gg 1$), where the responses are matrices instead of scalars, and we propose the following approaches to handle the problem.

3.1 | EE method

When $d > 1$, the simulation outputs $\{\mathbf{H}^h(\mathbf{x}_k), h = 1, \dots, n_k, k = 1, \dots, N\}$ are matrices, and the classical stochastic kriging cannot be applied directly. One natural way is to decompose the matrix by elements. That is, we treat each element in the matrix separately, and then use the stochastic kriging on each element to fit the surface. Let $H^h_{i,j}(\mathbf{x}_k)$ denote the (i, j)th element in the matrix $\mathbf{H}^h(\mathbf{x}_k)$, $h = 1, \dots, n_k$. Then, we use $\{\mathbf{x}_k, H^h_{i,j}(\mathbf{x}_k), h = 1, 2, \dots, n_k\}$, $k = 1, \dots, N$, to predict the (i, j)th element in $\mathbf{\Phi}(\mathbf{x}_0)$ via the stochastic kriging. Specifically, we assume that the hth simulation replication is modeled as

$$H^h_{i,j}(\mathbf{x}) = \beta_{i,j} + \mathsf{M}_{i,j}(\mathbf{x}) + \varepsilon^h_{i,j}(\mathbf{x}), \tag{3}$$

which is similar to (1). Let $\Sigma_{\mathsf{M}_{i,j}}$ denote the covariance matrix among the response values at design points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, with the (p, q)th element in the matrix $(\mathbf{\Sigma}_{\mathsf{M}_{i,j}})_{p,q} = \mathrm{Cov}(\mathsf{M}_{i,j}(\mathbf{x}_p), \mathsf{M}_{i,j}(\mathbf{x}_q)) = \tau^2_{i,j} R(\mathbf{x}_p, \mathbf{x}_q; \boldsymbol{\theta}_{i,j})$, where R is introduced in Section 2.2, and $\Sigma_{\mathsf{M}_{i,j}}(\mathbf{x}_0, \cdot)$ is the covariance vector with the qth element $(\Sigma_{\mathsf{M}_{i,j}}(\mathbf{x}_0, \cdot))_q = \mathrm{Cov}(\mathsf{M}_{i,j}(\mathbf{x}_0), \mathsf{M}_{i,j}(\mathbf{x}_q))$. Let

$$\mathbf{\Sigma}_{\varepsilon_{i,j}} = \mathrm{diag}[\mathrm{Var}(\overline{\varepsilon}_{i,j}(\mathbf{x}_1)), \mathrm{Var}(\overline{\varepsilon}_{i,j}(\mathbf{x}_2)), \dots, \mathrm{Var}(\overline{\varepsilon}_{i,j}(\mathbf{x}_N))],$$

where $\overline{\varepsilon}_{i,j}(\mathbf{x}_k) = 1/n_k \sum_{h=1}^{n_k} \varepsilon^h_{i,j}(\mathbf{x}_k)$, $k = 1, 2, \dots, N$. So the (i, j)th element in $\mathbf{\Phi}(\mathbf{x}_0)$, which denotes by $\Phi_{i,j}(\mathbf{x}_0)$, is predicted by

$$\widehat{\Phi}_{i,j}(\mathbf{x}_0) = \widehat{\beta}_{i,j} + \widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}}(\mathbf{x}_0, \cdot)^\top [\widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}} + \widehat{\mathbf{\Sigma}}_{\varepsilon_{i,j}}]^{-1} (\overline{\mathbf{H}}_{i,j} - \widehat{\beta}_{i,j} \mathbf{1}_N), \tag{4}$$

where $\overline{\mathbf{H}}_{i,j} = (\overline{H}_{i,j}(\mathbf{x}_1), \overline{H}_{i,j}(\mathbf{x}_2), \dots, \overline{H}_{i,j}(\mathbf{x}_N))^\top$ with $\overline{H}_{i,j}(\mathbf{x}_k) = 1/n_k \sum_{h=1}^{n_k} H^h_{i,j}(\mathbf{x}_k)$, $k = 1, 2, \dots, N$. As introduced in Section 2.1, the parameters $\widehat{\beta}_{i,j}$, $\widehat{\tau}^2_{i,j}$, and $\widehat{\boldsymbol{\theta}}_{i,j}$ are estimated via MLE, so that $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}}(\mathbf{x}_0, \cdot)$ and $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}}$ can be estimated accordingly, and $\widehat{\mathbf{\Sigma}}_{\varepsilon_{i,j}}$ is estimated by the sample variances. Notice that $\Phi(\mathbf{x}_0)$ is a symmetric matrix, so we only construct the upper (or lower) triangular element surfaces, and the total number of surfaces need to be fitted is $d(d+1)/2$. We call this approach the EE method, and we summarize the procedure in Algorithm 1.

Algorithm 1 provides a way to fit the gamma matrix surface, and we can use the EE method to solve the delta surface fitting problem as well. In the offline-simulation stage, we use this algorithm to build up the predictive model (gamma surface). In the online-application stage, the gamma matrix can be directly calculated by the predictive model. Then, it can be applied in real-time applications, for example, real-time delta-gamma hedging. However, the EE method has two drawbacks:

- The computational effort is high if d is large, because we need to fit $d(d+1)/2$ surfaces. In practice, financial portfolios may contain hundreds of assets, which makes the gamma matrix very large.
- The EE method considers all elements in isolation and ignores the relationship among them. Therefore, it may destroy the structure of the matrices and result in some severe problems. For example, we consider the gamma matrix of an arithmetic average basket call option with two underlying assets. In this setting, the responses are two-dimensional Hessian matrices (i.e., the gamma) of function $\mathbb{E}[Y(\mathbf{x})]$. Notice that $\mathbb{E}[Y(\mathbf{x})]$ is convex, therefore, the gamma matrices are positive semidefinite. However, if we

Algorithm 1 Gamma surface estimation via EE

Input: The design points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$; The simulation outputs of gamma matrices $\{\mathbf{H}^h(\mathbf{x}_k), h = 1, 2, \dots, n_k\}, k = 1, \dots, N$, with the (i, j)th element $H^h_{i,j}(\mathbf{x}_k)$; the new point $\mathbf{x}_0$.

1: **for** $i = 1$ to d **do**
2: **for** $j \geq i$ **do**
3: Estimate $\widehat{\mathbf{\Sigma}}_{\epsilon_{i,j}}$ via the sample variance. Based on the model (3), use MLE to estimate the parameters $\hat{\beta}_{i,j}$, $\hat{\tau}^2_{i,j}$, and $\hat{\boldsymbol{\theta}}_{i,j}$. Then calculate $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}}$ and $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_{i,j}}(\mathbf{x}_0, \cdot)$.
4: Use stochastic kriging to obtain the (i, j)th element surface in the gamma matrix. The predictor $\widehat{\Phi}_{i,j}(\mathbf{x}_0)$ on the new point $\mathbf{x}_0$ is given by (4).
5: Let $\widehat{\Phi}_{j,i}(\mathbf{x}_0) = \widehat{\Phi}_{i,j}(\mathbf{x}_0)$.
6: **end for** j
7: **end for** i

Output: Return the entire gamma surface $\widehat{\mathbf{\Phi}}(\mathbf{x}_0)$, where the (i, j)th element is $\widehat{\Phi}_{i,j}(\mathbf{x}_0)$.

decompose the matrix by elements, and fit the element surfaces separately, then the fitted matrices may not be positive semidefinite. Appendix A shows an example that one predicted matrix by the EE method at $\mathbf{x}_0 = [110.25; 110]$ is

$$\widehat{\mathbf{\Phi}}(\mathbf{x}_0) = \begin{pmatrix} 3.56 \times 10^{-3} & 3.96 \times 10^{-3} \\ 3.96 \times 10^{-3} & 3.16 \times 10^{-3} \end{pmatrix},$$

which has a negative eigenvalue -5.88×10^{-4}. We find that this is not a rare case in the normal setting, and this problem will be more severe if the pathwise estimators have large variances.

In the next two subsections, we propose two methods to tackle these two drawbacks.

3.2 | Importance mapping method

First, we consider how to overcome the drawback of high computational effort for multi-dimensional matrices. One way is to map the matrix to a scalar that summarizes the important property of the matrix. We construct an *importance mapping*

$$\Psi : \mathcal{S}^d \to \mathbb{R}$$

to represent the critical property of the gamma matrix. Recall that $\mathcal{S}^d$ is the space of symmetric matrices and $\mathbb{R}$ is the real line. Notice that there are many ways to construct the mapping Ψ. For instance, the operators, such as trace, minimum or maximum eigenvalue, and determinant, are all such mappings and they summarize the critical characteristics of the matrix in different ways.

For any design point $\mathbf{x}$, we construct the corresponding $\pi(\mathbf{x})$ via compound mappings $\Psi \cdot \mathbf{\Phi}$, such that $\pi(\mathbf{x}) = \Psi(\mathbf{\Phi}(\mathbf{x})) \in \mathbb{R}$. For any design point $\mathbf{x}_k$, let $\Pi^h(\mathbf{x}_k) \triangleq \Psi(\mathbf{H}^h(\mathbf{x}_k))$ be the hth simulation output after Ψ mapping on the matrix $\mathbf{H}^h(\mathbf{x})$ at the design point $\mathbf{x}_k$. Similar to (1), we model $\Pi^h(\mathbf{x})$ as

$$\Pi^h(\mathbf{x}) = \beta_{I0} + \mathsf{M}_I(\mathbf{x}) + \varepsilon^h_I(\mathbf{x}), \tag{5}$$

where β_{I0} is a constant. Let Σ_{M_I} denote the covariance matrix among $\mathsf{M}_I(\mathbf{x}_i)$ and $\mathsf{M}_I(\mathbf{x}_j)$, $i, j = 1, 2, \dots, N$, with the spatial parameter $\boldsymbol{\theta}_I$ and τ^2_I, and $\Sigma_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)$ denote the covariance vector among $\mathsf{M}_I(\mathbf{x}_0)$ and $\mathsf{M}(\mathbf{x}_i)$, $i = 1, 2, \dots, N$. The simulation noise matrix is denoted by

$$\mathbf{\Sigma}_{\varepsilon_I} = \mathrm{diag}[\mathrm{Var}(\overline{\varepsilon}_I(\mathbf{x}_1)), \mathrm{Var}(\overline{\varepsilon}_I(\mathbf{x}_2)), \dots, \mathrm{Var}(\overline{\varepsilon}_I(\mathbf{x}_N))],$$

where $\overline{\varepsilon}_I(\mathbf{x}_k) = 1/n_k \sum_{h=1}^{n_k} \varepsilon^h_I(\mathbf{x}_k), k = 1, 2, \dots, N$.

After determining the mapping Ψ and transferring the matrix data to the scalar data, we can construct the surface of $\pi(\mathbf{x})$ using stochastic kriging introduced in Section 2.2. Notice that the kriging is a kind of linear predictor, because the response value at $\mathbf{x}_0$ is predicted by a linear combination of the given response values at $\mathbf{x}_i$, $i = 1, 2, \dots, N$. Specifically, stochastic kriging uses the linear form $\omega(\mathbf{x}_0) + \boldsymbol{\omega}(\mathbf{x}_0)^\top \overline{\mathbf{\Pi}}$ to predict $\pi(\mathbf{x}_0)$, where $\overline{\mathbf{\Pi}} = (\overline{\Pi}(\mathbf{x}_1), \overline{\Pi}(\mathbf{x}_2), \dots, \overline{\Pi}(\mathbf{x}_N))^\top$ with $\overline{\Pi}(\mathbf{x}_k) = 1/n_k \sum_{h=1}^{n_k} \Pi^h(\mathbf{x}_k), k = 1, 2, \dots, N$, and $\omega^*(\mathbf{x}_0) = \beta_{I0} - \beta_{I0} \mathbf{\Sigma}\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \mathbf{1}_N$ and $\boldsymbol{\omega}^*(\mathbf{x}_0)^\top = \mathbf{\Sigma}\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1}$. If the optimal $\omega^*(\mathbf{x}_0)$ and $\boldsymbol{\omega}^*(\mathbf{x}_0)$ are obtained, we can then use the same weight to each element in the matrix to obtain the element surfaces, that is, the (i, j)th element in gamma matrix at $\mathbf{x}_0$ can be predicted by $\omega^*(\mathbf{x}_0) + \boldsymbol{\omega}^*(\mathbf{x}_0)^\top \overline{\mathbf{H}}_{i,j}$, $i, j = 1, 2, \dots, d$. Therefore, we build the gamma surface.

However, in this approach, the predictor $\omega^*(\mathbf{x}_0) + \boldsymbol{\omega}^*(\mathbf{x}_0)^\top \overline{\mathbf{H}}_{i,j}$ may be biased from the true value $\beta_{i,j}$, $i, j = 1, 2, \dots, d$. Specifically, since $\mathbb{E}[\overline{\mathbf{H}}_{i,j}] = \beta_{i,j} \mathbf{1}_N$,

$$\begin{aligned} &\mathbb{E}[\omega^*(\mathbf{x}_0) + \boldsymbol{\omega}^*(\mathbf{x}_0)^\top \overline{\mathbf{H}}_{i,j}] \\ &\quad = \mathbb{E}[\beta_{I0} - \beta_{I0} \mathbf{\Sigma}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}_\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \mathbf{1}_N \\ &\qquad + \mathbf{\Sigma}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}_\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \overline{\mathbf{H}}_{i,j}] \\ &\quad = \beta_{I0}(1 - \mathbf{\Sigma}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}_\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \mathbf{1}_N) \\ &\qquad + \beta_{i,j} \mathbf{\Sigma}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}_\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \mathbf{1}_N. \end{aligned} \tag{6}$$

Notice that $\mathbf{\Sigma}\mathsf{M}(\mathbf{x}_0, \cdot)^\top [\mathbf{\Sigma}\mathsf{M} + \mathbf{\Sigma}_\varepsilon]^{-1} \mathbf{1}_N$ does not necessarily equal to 1, so the predictor is biased. One possible way to overcome this problem is to replace β_{I0} in $\omega^*(\mathbf{x}_0)$ by $\beta_{i,j}$, then by (6), it is easy to show that the predictor is unbiased. However, this way needs to estimate $\beta_{i,j}$ in each element model (3), where we need to implement the MLE $d(d+1)/2$ times to estimate $\beta_{i,j}$. As a better way to address

the computational difficulty in calculating many MLEs, we consider another kriging method in deterministic simulation experiments, called *ordinary kriging*, which does not necessarily know $\beta_{i,j}$, see Stein (1999) and van Beers and Kleijnen (2003) considered to use the same kriging method in stochastic simulation. To distinguish from the stochastic kriging method introduced in Section 2.2, we call this approach *stochastic ordinary kriging*.

Specifically, we assume that the $\pi(\mathbf{x}_0) = \beta_{I0} + \mathsf{M}_I(\mathbf{x}_0)$ is predicted by the linear predictor

$$\mathbf{w}(\mathbf{x}_0)^\top \overline{\mathbf{\Pi}} = \sum_{i=1}^{N} w_i(\mathbf{x}_0)\overline{\Pi}(\mathbf{x}_i), \text{where } \sum_{i=1}^{N} w_i(\mathbf{x}_0) = 1,$$

where $\mathbf{w}(\mathbf{x}_0) = (w_1(\mathbf{x}_0), w_2(\mathbf{x}_0), \ldots, w_N(\mathbf{x}_0))$. Similar to ordinary kriging (see Cressie, 1993), the criterion to choose the optimal $\mathbf{w}^*(\mathbf{x}_0)$ is to minimize the mean squared prediction error defined as $\mathbb{E}\left[\left(\pi(\mathbf{x}_0) - \sum_{i=1}^{N} w_i(\mathbf{x}_0)\overline{\Pi}(\mathbf{x}_i)\right)^2\right]$ subject to $\sum_{i=1}^{N} w_i(\mathbf{x}_0) = 1$. Then, by Lagrangian relaxation, we need to solve

$$\min_{\mathbf{w}} \mathbb{E}\left[\left(\pi(\mathbf{x}_0) - \sum_{i=1}^{N} w_i(\mathbf{x}_0)\overline{\Pi}(\mathbf{x}_i)\right)^2\right] + 2\alpha\left(\sum_{i=1}^{N} w_i(\mathbf{x}_0) - 1\right), \quad (7)$$

where α is the Lagrangian multiplier. Notice that $(\Sigma_{\mathsf{M}_I})_{i,j} = \mathrm{Cov}(\mathsf{M}_I(\mathbf{x}_i), \mathsf{M}_I(\mathbf{x}_j))$ and $(\mathbf{\Sigma}_{\varepsilon_I})_{i,i} = \mathrm{Var}\left(1/n_i \sum_{h=1}^{n_i} \varepsilon_I^h(\mathbf{x}_i)\right)$, and $(\mathbf{\Sigma}_{\varepsilon_I})_{i,j} = 0$ for $i \neq j$. Then, (7) is equivalent to

$$\min_{\mathbf{w}} \sum_{i=1}^{N}\sum_{j=1}^{N} w_i(\mathbf{x}_0)w_j(\mathbf{x}_0)((\mathbf{\Sigma}_{\mathsf{M}_I})_{i,j} + (\mathbf{\Sigma}_{\varepsilon_I})_{i,j}) - 2\sum_{i=1}^{N} w_i(\mathbf{x}_0)(\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot))_i + 2\alpha\left(\sum_{i=1}^{N} w_i(\mathbf{x}_0) - 1\right).$$

According to van Beers and Kleijnen (2003), the optimal $\mathbf{w}^*(\mathbf{x}_0)$ is given by

$$\mathbf{w}^*(\mathbf{x}_0)^\top = \left(\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot) + \mathbf{1}_N \frac{1 - \mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)}{\mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}.$$

Thus, the predictor is given by

$$\hat{\pi}(\mathbf{x}_0) = \left(\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot) + \mathbf{1}_N \frac{1 - \mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)}{\mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\mathbf{\Sigma}_{\mathsf{M}_I} + \mathbf{\Sigma}_{\varepsilon_I})^{-1}\overline{\Pi}(\mathbf{x}_i).$$

$\mathbf{\Sigma}_{\varepsilon_I}$ can be estimated using the sample variance, that is, let

$$\hat{V}(\mathbf{x}_i) = \frac{1}{n_i - 1}\sum_{h=1}^{n_i} (\Pi^h(\mathbf{x}_i) - \overline{\Pi}(\mathbf{x}_i))^2, \quad (8)$$

then

$$\hat{\mathbf{\Sigma}}_{\varepsilon_I} = \mathrm{diag}\{\hat{V}(\mathbf{x}_1)/n_1, \hat{V}(\mathbf{x}_2)/n_2, \ldots, \hat{V}(\mathbf{x}_N)/n_N\}. \quad (9)$$

Similar to Theorem 1 of Ankenman et al. (2010), we make the following assumption, and then prove that estimating $\mathbf{\Sigma}_{\varepsilon_I}$ by the sample variance introduces no prediction bias. The proof is included in Appendix B.

Assumption 1 *Suppose that $\Pi^h(\boldsymbol{x})$ is modeled by* (5). *M_I is second-order stationary Gaussian random field, and $\varepsilon_I^1(\mathbf{x}_i), \varepsilon_I^2(\mathbf{x}_i), \ldots, \varepsilon_I^{n_i}(\mathbf{x}_i)$ are i.i.d. normal distribution with mean* 0 *and variance $V(\boldsymbol{x}_i)$, independent of $\varepsilon^h(\boldsymbol{x}_j)$ for all h and $j \neq i$, and independent of M_I.*

Theorem 1 *Under Assumption* 1, *suppose that $\hat{\mathbf{\Sigma}}_{\varepsilon_I}$ is given by* (9), *and let*

$$\hat{\hat{\pi}}(\mathbf{x}_0) = \left(\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot) + \mathbf{1}_N \frac{1 - \mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)}{\mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}\overline{\mathbf{\Pi}}.$$

Then, $\mathbb{E}[\hat{\hat{\pi}}(\mathbf{x}_0) - \pi(\mathbf{x}_0)] = 0$.

After obtaining the optimal weights of the surface of $\pi(\mathbf{x}_0)$, we can apply the weights directly into each element of the gamma matrix. Let

$$\hat{\mathbf{w}}^*(\mathbf{x}_0)^\top = \left(\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot) + \mathbf{1}_N \frac{1 - \mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}\mathbf{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)}{\mathbf{1}_N^\top(\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\mathbf{\Sigma}_{\mathsf{M}_I} + \hat{\mathbf{\Sigma}}_{\varepsilon_I})^{-1}. \quad (10)$$

Then

$$\hat{\mathbf{\Phi}}_{i,j}(\mathbf{x}_0) = \hat{\mathbf{w}}^*(\mathbf{x}_0)^\top \overline{\mathbf{H}}_{i,j}, \text{for } i \leq j, i, j = 1, 2, \ldots, d, \quad (11)$$

that is, to obtain the surface of the element $\Phi_{i,j}(\mathbf{x}_0)$ in the matrix, we use the same weight of the surface of $\pi(\mathbf{x}_0)$. We call this approach the *importance mapping* (*IM*) method, and we summarize the procedure in Algorithm 2. This algorithm shows the procedure of building up the predictive model via IM in the offline-simulation stage. In the online-application stage, we use the predictive model to do the real-time delta-gamma hedging.

Under the appropriate assumptions, we can prove that the predictor of each element in the matrix is unbiased. The proof is included in Appendix C.

Assumption 2 *Suppose that each of the elements in the matrix is modeled by* (3). *For given i, j, $\mathsf{M}_{i,j}$ is second-order stationary Gaussian random field, and $\varepsilon_{i,j}^1(\mathbf{x}_k), \varepsilon_{i,j}^2(\mathbf{x}_k), \ldots, \varepsilon_{i,j}^{n_i}(\mathbf{x}_k)$ are i.i.d. normal distribution with mean* 0 *and variance $V_{i,j}(\boldsymbol{x}_k)$, independent of $\varepsilon_{i,j}^h(\mathbf{x}_p)$ for all h and $p \neq k$, and independent of $\mathsf{M}_{i,j}$.*

Theorem 2 *Under Assumptions* 1 and 2, *suppose that $\hat{\mathbf{\Sigma}}_{\varepsilon_I}$ is given by* (9), *and the predictor $\hat{\mathbf{\Phi}}_{i,j}(\mathbf{x}_0)$ is given by* (11). *For any given linear*

Algorithm 2 Gamma surface estimation via IM

Input: The design points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$; The simulation outputs of gamma matrices $\{\mathbf{H}^h(\mathbf{x}_k), h = 1, 2, \dots, n_k\}, k = 1, 2, \dots, N$; the new point $\mathbf{x}_0$.

1: Map the gamma matrices to scalars $\Pi^h(\mathbf{x}_k) = \Psi(\mathbf{H}^h(\mathbf{x}_k))$, $h = 1, 2, \dots, n_k, k = 1, 2, \dots, N$.
2: Estimate $\widehat{\mathbf{\Sigma}}_{\epsilon_I}$ via the sample variance. Use MLE to estimate the parameters $\hat{\boldsymbol{\theta}}_I$ and $\hat{\tau}_I^2$, then calculate $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_I}$ and $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)$.
3: Use stochastic ordinary kriging to obtain the surface of $\pi(\mathbf{x}_0)$. The optimal weight $\widehat{\mathbf{w}}^*(\mathbf{x}_0)$ of the linear combination is given by (10) with substituting $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_I}$ and $\widehat{\mathbf{\Sigma}}_{\mathsf{M}_I}(\mathbf{x}_0, \cdot)$.
4: **for** $i = 1$ to d **do**
5: **for** $j \geq i$ **do**
6: Let $\widehat{\Phi}_{i,j}(\mathbf{x}_0) = \widehat{\mathbf{w}}^*(\mathbf{x}_0)^\top \bar{\mathbf{H}}_{i,j}$, and $\widehat{\Phi}_{j,i}(\mathbf{x}_0) = \widehat{\Phi}_{i,j}(\mathbf{x}_0)$.
7: **end for** j
8: **end for** i

Output: Return the entire gamma surface $\widehat{\mathbf{\Phi}}(\mathbf{x}_0)$, where the (i, j)th element in it is $\widehat{\Phi}_{i,j}(\mathbf{x}_0)$.

importance mapping Ψ *such that* $\Psi(\mathbf{\Phi}(\mathbf{x})) = \sum_{i=1}^d \sum_{j=1}^d u_{i,j}\Phi_{i,j}(\mathbf{x})$, *where* $u_{i,j}$, $i, j = 1, 2, \dots, d$, *are the weights in the linear mapping, we have* $\mathbb{E}[\widehat{\Phi}_{i,j}(\mathbf{x}_0) - \Phi_{i,j}(\mathbf{x}_0)] = 0$ *for* $i, j = 1, 2, \dots, d$.

Notice that the trace operator is a linear mapping, where $u_{i,j} = 1$ for $i = j$, and $u_{i,j} = 0$ for $i \neq j$. Here, the linear importance mapping Ψ is required, because it is a sufficient condition to achieve the independence between the sample variance of simulation noise $\widehat{V}_{i,j}(\mathbf{x}_k)$ at each design point $\mathbf{x}_k$ and the sample mean $\overline{H}_{i,j}$ in Algorithm 2. So we can prove that the importance mapping method is still unbiased along with Theorem 1. But for nonlinear importance mapping such as maximum eigenvalue and determinant, the sample variance and the sample mean of simulation noise may not be independent, so the importance mapping method may introduce bias in fitting gamma surface in general.

3.3 | Cholesky decomposition

When we consider the gamma of derivatives with multiple underlying assets, that is, the Hessian matrix of the derivative price with respect to underlying asset prices, it may have certain properties. If the derivative price is convex with respect to underlying asset prices, the corresponding gamma matrix is positive semidefinite. However, the EE method cannot maintain this property (see the example in Section 2.1). For the IM method, the component $\widehat{w}_i(\mathbf{x}_0), i = 1, 2, \dots, N$, in $\widehat{\mathbf{w}}^*(\mathbf{x}_0)$ may be negative, so the estimated gamma matrix via the IM method does not guarantee to be positive semidefinite either. In this subsection, we propose a new approach to maintain the positive (semi)definiteness of the gamma matrix when this property is important.

The Cholesky decomposition or Cholesky factorization is a decomposition of a positive-definite matrix into the product of a unique lower triangular matrix and its transpose; see Meyer (2013). Specifically, if the derivative price is strictly convex with respect to the underlying assets price, the gamma matrix $\mathbf{\Phi}(\mathbf{x})$ is positive definite, but not for each simulation output $\mathbf{H}^h(\mathbf{x}_k)$ calculated by SPA estimator, which is the Hessian matrix of conditional expectation of payoff function $\mathbb{E}[Y(\boldsymbol{\theta})|Z]$. The realization of the Hessian matrix on each sample path may not be positive definite although its expectation is positive definite. Some examples are shown in Appendix D. Here, we can assume that the sample mean of the simulation output $\overline{\mathbf{H}}(\mathbf{x}_k) = 1/n_k \sum_{h=1}^{n_k} \mathbf{H}^h(\mathbf{x}_k)$ is positive definite by choosing larger n_k, since it converges to $\mathbf{\Phi}(\mathbf{x}_k)$ as $n_k \to \infty$ with probability 1. So we can apply Cholesky decomposition to $\overline{\mathbf{H}}(\mathbf{x}_k)$ to obtain a lower triangular matrix $\overline{\mathbf{L}}(\mathbf{x}_k)$, such that

$$\overline{\mathbf{H}}(\mathbf{x}_k) = \overline{\mathbf{L}}(\mathbf{x}_k)\overline{\mathbf{L}}(\mathbf{x}_k)^\top.$$

For the gamma matrix $\mathbf{\Phi}(\mathbf{x}_0)$, let $\boldsymbol{\phi}(\mathbf{x}_0)$ be the lower triangular matrix of the Cholesky decomposition, that is

$$\mathbf{\Phi}(\mathbf{x}_0) = \boldsymbol{\phi}(\mathbf{x}_0)\boldsymbol{\phi}(\mathbf{x}_0)^\top.$$

Therefore, instead of predicting $\mathbf{\Phi}(\mathbf{x}_0)$, we can predict the lower triangular matrix $\boldsymbol{\phi}(\mathbf{x}_0)$ first, whose predictor is denoted by $\widehat{\boldsymbol{\phi}}(\mathbf{x}_0)$, then $\mathbf{\Phi}(\mathbf{x}_0)$ is predicted by

$$\widehat{\mathbf{\Phi}}(\mathbf{x}_0) = \widehat{\boldsymbol{\phi}}(\mathbf{x}_0)\widehat{\boldsymbol{\phi}}(\mathbf{x}_0)^\top.$$

To predict $\boldsymbol{\phi}(\mathbf{x}_0)$, we can still use the EE or IM method. If the dimension d is small, we may use the EE method (denoted by Chol-EE). If d is large, we may use the IM method (denoted by Chol-IM). Under both the EE and IM methods with a linear mapping, the predictors for $\boldsymbol{\phi}(\mathbf{x}_0)$ are still unbiased. However, the estimator of $\mathbf{\Phi}(\mathbf{x}_0)$ may not be unbiased because we construct the gamma by $\widehat{\boldsymbol{\phi}}(\mathbf{x}_0)\widehat{\boldsymbol{\phi}}(\mathbf{x}_0)^\top$. For further extension, we can apply LDL decomposition (see Meyer, 2013) instead of Cholesky decomposition to guarantee the positive semi-definiteness instead of positive definiteness.

There is another issue needs to be addressed in implementing Cholesky decomposition. We know that the simulation output $\mathbf{H}^h(\mathbf{x}_k)$ may sometimes not be positive definite. To be specific, Cholesky decomposition can only be conducted on a sample mean $\overline{\mathbf{H}}(\mathbf{x}_k)$ by increasing the sample size n_k. So we cannot estimate the covariance matrix of the simulation noise $\widehat{\mathbf{\Sigma}}_{\epsilon_{i,j}}$ for EE or $\widehat{\mathbf{\Sigma}}_{\epsilon_I}$ for IM via the sample variance directly (see (2) or (8)), since we cannot even conduct the Cholesky decomposition to obtain triangular matrices for those nonpositive semidefinite observations. Here we propose

to use bootstrap to estimate the covariance matrix of the simulation noise to overcome this difficulty. For more detail about bootstrap, refer to Efron and Tibshirani (1993). We provide the following procedure to approximate the variance of each element of $\overline{\mathbf{L}}(\mathbf{x}_k)$ via bootstrap:

(1) For given $\mathbf{x}_k$, draw n_k observations, with replacement, from the original data $\{\mathbf{H}^h(\mathbf{x}_k), h = 1, 2, \ldots, n_k\}$, and called these observations the bootstrap sample.
(2) Calculate the sample mean $\overline{\mathbf{H}}_b(\mathbf{x}_k)$ of the bootstrap sample, then apply Cholesky decomposition in order to obtain the corresponding low triangular matrix $\overline{\mathbf{L}}_b(\mathbf{x}_k)$.
(3) Implement the first two steps B times and obtain $\{\overline{\mathbf{L}}_b(\mathbf{x}_k), b = 1, 2, \ldots, B\}$.
(4) Let $\overline{\mathbf{L}}_b(\mathbf{x}_k)_{i,j}$ denote the (i, j)th element in the matrix $\overline{\mathbf{L}}_b(\mathbf{x}_k)$. Then the variance of the (i, j)th element of matrix $\overline{\mathbf{L}}(\mathbf{x}_k)$, denoted by $\mathrm{Var}(\overline{\mathbf{L}}(\mathbf{x}_k)_{i,j})$, is approximated by

$$\frac{1}{B}\sum_{b=1}^{B}\left[\overline{\mathbf{L}}_b(\mathbf{x}_k)_{i,j} - \frac{1}{B}\sum_{b=1}^{B}\overline{\mathbf{L}}_b(\mathbf{x}_k)_{i,j}\right]^2.$$

Here, n_k observations are drawn with replacement in bootstrap because we initially choose the large enough n_k to guarantee the positive semidefiniteness of $\overline{\mathbf{H}}(\mathbf{x}_k)$ as well as the positive semidefiniteness of $\overline{\mathbf{H}}_b(\mathbf{x}_k)$. After obtaining the variances of elements in $\overline{\mathbf{L}}(\mathbf{x}_k), k = 1, 2, \ldots, k$, we can apply Algorithms 1 and 2 to implement EE-Chol and IM-Chol, respectively.

To sum up, for fitting the gamma matrix surface, we propose three different types of approaches. The pros and cons are illustrated as follows:

- The first type is the EE method, that is, we first treat each element in the matrix separately as a single surface fitting problem, and then put all the element surfaces together. This type of methods is straightforward and easy to implement. However, the estimation accuracy is hard to guarantee. The assumption of this type of methods excludes the situation that the elements in one matrix are usually correlated.
- The second type is the IM method, that is, we first use an importance mapping to map the matrix to a scalar, and then apply the same linear combination weights in the scalar surface fitting to elements in the matrix to obtain the element surfaces. This type of methods can obtain the unbiased estimation and become really efficient if the dimension is quite large.
- The third type is the Cholesky decomposition-based method, that is, we first decompose the gamma matrix into a lower triangular matrix, and then use the EE or IM method to fit the element surfaces of the lower triangular matrix. Cholesky decomposition-based method can keep the positive definiteness of the gamma matrix.

4 | NUMERICAL EXPERIMENTS

In this section, we consider two derivatives with multi-underlying assets, the geometric average basket option and the arithmetic average basket option, as the representative examples to evaluate our proposed methods. The basket option is a family of options with a basket of underlying assets, such as securities, currencies, and so on. It gradually becomes quite popular in the financial market to hedge risk from the co-movement of several underlying assets, multinational exchange rates, and so on. Moreover, investors often prefer basket options to individual options on each underlying that makes up the basket, because they are cheaper.

We compare the proposed methods. In the IM methods, we use trace, maximum eigenvalue, determinant as the importance mappings. As discussed above, as the dimension of the response matrix increases, the computational burden of the EE and Chol-EE increases fast in the order of d^2, because they repeatedly conduct the MLE in the stochastic kriging procedure. Therefore, the IM method outperforms EE method in terms of computational effort. Obviously, Cholesky decomposition-based methods outperform those methods without Cholesky decomposition if we need to carefully consider the positive definiteness of estimated gamma matrices. In the following numerical experiments, we compare the estimation accuracy of all the methods. Specifically, we evaluate the performance of elements, trace, maximum eigenvalue, and minimum eigenvalue of the estimated matrix, and we further examine the performance of the proposed methods using the accuracy of the delta-gamma approximation in order to investigate the performance when gamma is used in risk management problems.

4.1 | Gamma estimation

Consider a vanilla basket call option with d underlying stocks $S_1(t), S_2(t), \ldots, S_d(t)$, where $S_i(t) = S_i(0)\exp((\mu_i - \sigma_i^2/2)t + \sigma_i W_i(t))$ is a geometric Brownian motion. Suppose that $S_1(t)$, $S_2(t), \ldots, S_d(t)$ are independent of each other. Let T be the maturity of the option, K be the strike price, and r be risk-free interest rate. The price of an arithmetic average basket call option is

$$\mathbb{E}[Y^A(\mathbf{x})] = e^{-rT}\mathbb{E}\left[\left(\frac{1}{d}\sum_{i=1}^{d}S_i(T) - K\right)^+\right],$$

where $\mathbf{x} = (S_1(0), S_2(0), \ldots, S_d(0))^\top$ is a vector containing all the initial underlying stock values and $Y^A(\mathbf{x})$ is the discounted payoff function.

The price of a geometric average basket call option is

$$\mathbb{E}[Y^G(\mathbf{x})] = e^{-rT}\mathbb{E}\left[\left(\sqrt[d]{S_1(T)S_2(T)\ \ldots\ S_d(T)} - K\right)^+\right],$$

where $Y^G(\mathbf{x})$ is the discounted payoff function. The gamma matrices of these call basket options

$$\mathbf{\Phi}^A(\mathbf{x}) = \nabla^2_{\mathbf{x}}\mathbb{E}[Y^A(\mathbf{x})],$$

and

$$\mathbf{\Phi}^G(\mathbf{x}) = \nabla^2_{\mathbf{x}}\mathbb{E}[Y^G(\mathbf{x})],$$

are two symmetric matrices with dimension d. Specifically, the expected payoff of arithmetic average basket call option is a convex function, then the gamma matrix of the basket option is positive semidefinite. Since $\mathbf{\Phi}^A(\mathbf{x})$ has zero minimum eigenvalues only in an area with probability measure zero, we can further claim that $\mathbf{\Phi}^A(\mathbf{x})$ is strictly positive definite.

For the selection of the design points $\{\mathbf{x}_k, k = 1, 2, \ldots, N\}$, we recommend Latin hypercube sampling (see Glasserman, 2013). On each design point $\mathbf{x}_k$, we apply SPA to estimate the gamma matrix. The SPA estimator for geometric average basket call option and arithmetic average basket call option are provided in Appendices E and F, respectively. Therefore, once we have the design points and the corresponding observations of estimated gamma matrices, we apply the proposed methods to fit the overall matrix surface and evaluate the performance at preset testing points.

Notice that the price of the geometric average basket option has an analytical formula, so the gamma matrix also has an analytical formula, which is provided in Appendix G. However, the price of arithmetic average call basket option is rather complicated, and the analytical formula is unavailable. Therefore, we evaluate the true gamma matrix by 10^8 samples at each testing point.

4.2 | Parameter settings

Let both the arithmetic average basket option and geometric average basket option share the same benchmark underlying stock values $\mathbf{S} = (S_1, S_2, \ldots, S_d)$, risk-free interest rate r, maturity T, strike price K, and volatility $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_d)$. The benchmark underlying stock value $\mathbf{S}$ is used to generate the design points. That is, the design point $\mathbf{x}_k \in [0.9S_1, 1.1S_1] \times [0.9S_2, 1.1S_2] \times \cdots \times [0.9S_d, 1.1S_d] \subset \mathbb{R}^d$ is selected by Latin hypercube sampling. For each design point $\mathbf{x}_k$, let the number of observations that estimate the gamma matrices be constant, that is, $n_k = n$ for all $k = 1, 2, \ldots, N$. The parameter setting in the experiments is listed in Table 1.[2] The testing points $\{\mathbf{x}^t_m, m = 1, 2, \ldots, N_t\}$ are also selected by Latin hypercube sampling in region $[0.95S_1, 1.05S_1] \times [0.95S_2, 1.05S_2] \times \cdots \times [0.95S_d, 1.05S_d] \subset \mathbb{R}^d$. Here, we set $N_t = 20$.

TABLE 1 Parameter setting

	$d = 2$	$d = 4$	$d = 6$
S	(105,100)	(105,100,95,105)	(106,95,103,102,100,105)
σ	(0.3,0.4)	(0.3,0.4,0.3,0.2)	(0.3,0.4,0.2,0.3,0.2,0.3)
n	1000/100	5000/500	20,000/2000
N	20	30	40
T		1	
r		0.03	
K		100	

[2]This table summarizes all the parameters for geometric average and arithmetic average basket option. In most of the cases, these two basket options share the same parameters except the sample size n. The sample sizes n that we need for geometric average and arithmetic average basket options are different. The number before the slash is for the arithmetic average basket option, and the number after the slash is for the geometric average basket option. In the arithmetic average basket option, n is larger because we need the sample mean of the gamma matrix to be positive definite.

4.3 | Experiment results

First, we compare the EE and IM methods. Specifically, for IM, we consider three matrix operators, trace (denoted by IM-tr), maximum eigenvalue (denoted by IM-eig), and determinant (denoted by IM-det). Consider the geometric average basket call option with two underlying assets ($d = 2$), and fix a testing point $\mathbf{x}^t = \mathbf{S} = (105, 100)$. We replicate the experiment $R = 20$ times, then plot the results in Figure 1. It shows the boxplot of the elements, trace, and eigenvalue of the gamma estimation results. The solid black line is the true value we want to estimate. Four methods are compared: EE, IM-tr, IM-eig, and IM-det. This figure indicates that all the methods perform well in estimating the gamma matrix.

Then, we consider $d = 4$, and compare the performances of these methods on the given testing points. For a summary presentation of the results, we calculate the relative bias (rBias), the relative standard deviation (rSD), and the relative root mean square error (rRMSE) of the prediction over the testing points, which are defined as (for the (i, j)th element)

$$rBias(i,j) = \frac{1}{N_t R}\sum_{m=1}^{N_t}\sum_{r=1}^{R}\left|\frac{\widehat{\Phi}^r_{i,j}(\mathbf{x}^t_m) - \Phi_{i,j}(\mathbf{x}^t_m)}{\Phi_{i,j}(\mathbf{x}^t_m)}\right|,$$

$$rSD(i,j) = \frac{1}{N_t}\sum_{m=1}^{N_t}\frac{\sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(\widehat{\Phi}^r_{i,j}(\mathbf{x}^t_m) - \frac{1}{R}\sum_{r=1}^{R}\widehat{\Phi}^r_{i,j}(\mathbf{x}^t_m)\right)^2}}{|\Phi_{i,j}(\mathbf{x}^t_m)|},$$

$$rRMSE(i,j) = \frac{1}{N_t}\sum_{m=1}^{N_t}\frac{\sqrt{\frac{1}{R}\sum_{r=1}^{R}(\widehat{\Phi}^r_{i,j}(\mathbf{x}^t_m) - \Phi_{i,j}(\mathbf{x}^t_m))^2}}{|\Phi_{i,j}(\mathbf{x}^t_m)|},$$

where $\widehat{\Phi}^r_{i,j}(\mathbf{x}^t_m)$ is the predicted gamma at the mth testing point in the rth replication of the experiment. The gamma estimation performance of overall surfaces measured by rBias, rSD, and rRMSE. Table 2 shows the element estimation performance of geometric average gamma matrices with four underlying assets by replicating the experiment 20 times. Testing points are randomly selected by Latin hypercube sampling. According to rRMSE, the top performance is marked as bold comparing among four methods: EE, IM-tr, IM-eig,

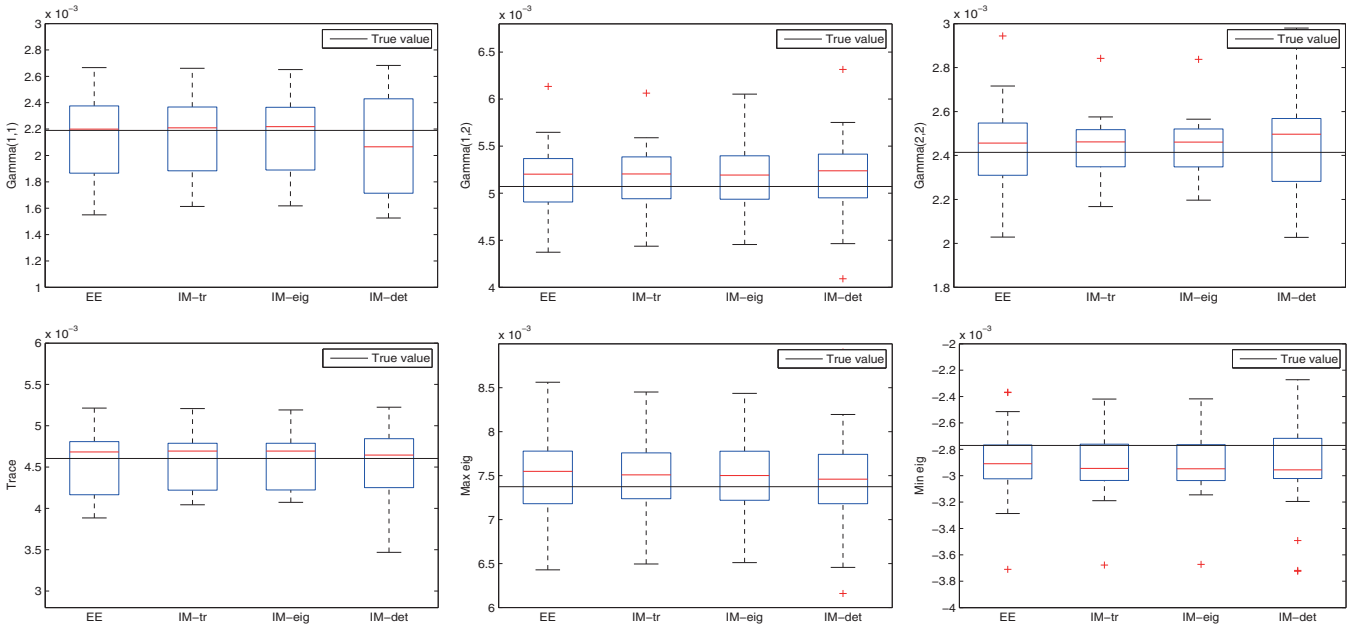

FIGURE 1 Boxplots of the gamma estimation in the geometric average basket option ($d = 2$). EE, element-by-element; IM, importance mapping [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Gamma estimation of geometric average call basket option with four underlying assets

	rBias (%)				rSD (%)				rRMSE (%)			
	EE	**IM-tr**	**IM-eig**	**IM-det**	**EE**	**IM-tr**	**IM-eig**	**IM-det**	**EE**	**IM-tr**	**IM-eig**	**IM-det**
(1,1)	9.62	9.62	9.43	9.64	11.05	10.97	10.96	10.98	14.65	**14.60**	14.47	14.62
(2,1)	2.74	2.74	2.75	2.74	2.70	2.67	2.70	2.66	3.87	**3.84**	3.88	3.84
(2,2)	10.28	10.25	10.19	10.29	10.07	10.07	10.05	10.08	14.47	14.44	**14.39**	14.48
(3,1)	2.68	2.68	2.68	2.67	2.60	2.57	2.60	2.54	3.74	3.72	3.74	**3.69**
(3,2)	3.24	3.23	3.21	3.24	1.94	1.94	1.92	1.94	3.79	3.79	**3.76**	3.79
(3,3)	9.66	9.67	9.57	9.67	11.27	11.28	11.21	11.26	14.87	14.88	**14.76**	14.86
(4,1)	2.74	2.74	2.76	2.74	2.72	2.69	2.72	2.69	3.87	**3.85**	3.88	3.85
(4,2)	3.19	3.19	3.18	3.19	1.84	1.84	1.80	1.84	3.70	3.70	**3.67**	3.70
(4,3)	2.93	2.94	2.92	2.95	2.62	2.61	2.60	2.62	3.94	3.95	**3.92**	3.96
(4,4)	14.87	14.85	14.60	14.87	18.30	18.24	18.16	18.29	23.59	23.53	**23.31**	23.59

Abbreviations: EE, element-by-element; IM, importance mapping; rBias, relative bias; rRMSE, relative root mean square error; rSD, relative standard deviation.

and IM-det. This table shows that the rBias, rSD, and rRMSE of all the proposed methods are small in most cases. And the larger estimation error for diagonal entries of gamma may result from the fact that their true values are relatively close to zero in this setting. Notice that EE costs a long computation time since it needs to conduct MLEs and stochastic kriging procedures many times. So based on the results in this experiment, the IM-based methods are more appealing in fitting the gamma matrix surface.

Next, we consider the arithmetic average basket call option, and compare EE, IM-tr, IM-eig, IM-det, and the methods with Cholesky decomposition. Specifically, we consider Chol-EE (Cholesky decomposition with EE), Chol-tr (Cholesky decomposition with IM via trace), Chol-eig (Cholesky decomposition with IM via maximum eigenvalue), and Chol-det (Cholesky decomposition with IM via determinant). Similar to the geometric average basket option, we first let $d = 2$, and fix a testing point $\mathbf{x}^t = \mathbf{S} = (105, 100)$. We replicate the experiment $R = 20$ times, then plot the results in Figure 2. This figure shows the boxplot of the elements, trace, and eigenvalue of the gamma estimation results. The solid black line is the true value we want to estimate. Six methods are compared: EE, IM-tr, Chol-EE, Chol-tr, Chol-eig, and Chol-det. This figure indicates that all the methods can provide good estimation of the gamma matrix.

Then, we consider $d = 4$ and $d = 6$. The results are still presented in terms of rBias, rSD, and rRMSE on all the testing points in Tables 3 and 4. Tables 3 and 4 show the element estimation performance of arithmetic average gamma matrices with four and six underlying assets by replicating the experiment 20 times. According to rRMSE, the top performance is marked as bold comparing among six methods: EE, IM-tr, Chol-EE, Chol-tr, Chol-eig, Chol-det. Similar to the

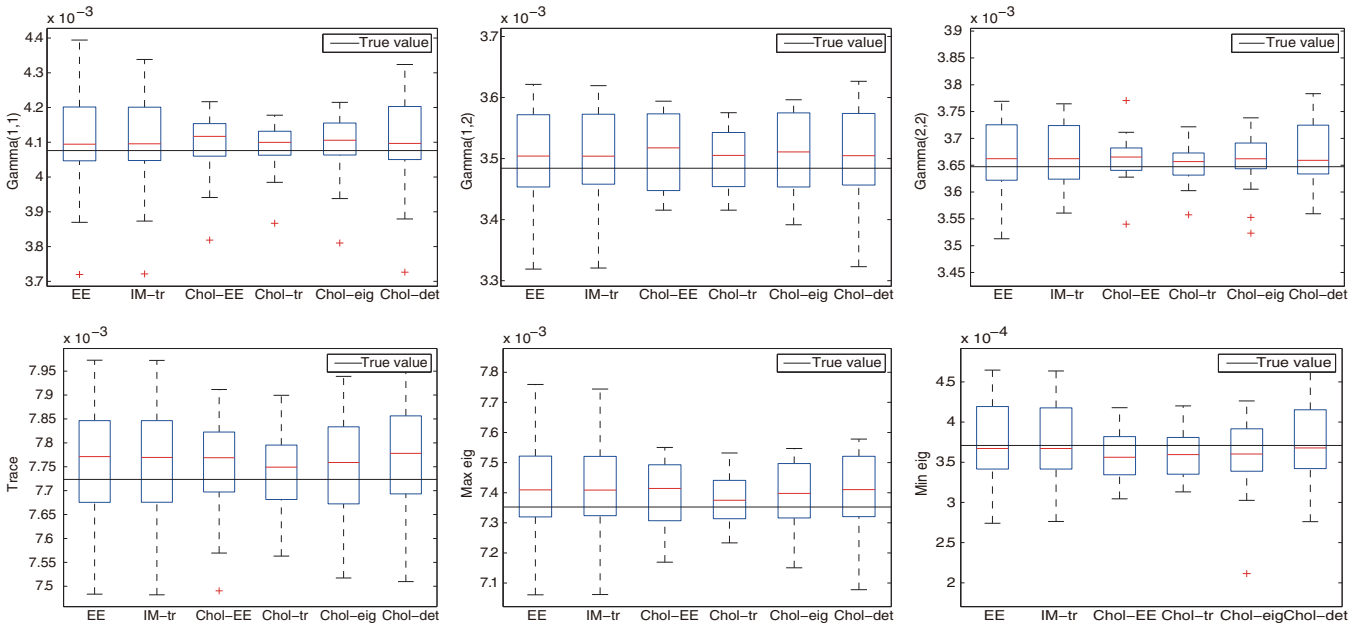

FIGURE 2 Boxplots of gamma estimation in the arithmetic-average basket option ($d = 2$). EE, element-by-element; IM, importance mapping [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Gamma estimation of arithmetic-mean call basket option with four underlying assets

	rBias (%)			rSD (%)			rRMSE (%)		
	EE	**IM-tr**	**Chol-EE**	**EE**	**IM-tr**	**Chol-EE**	**EE**	**IM-tr**	**Chol-EE**
(1,1)	1.97	1.97	1.95	2.60	2.61	2.57	3.27	3.28	3.22
(2,1)	2.66	2.46	2.27	3.56	3.37	3.04	4.45	4.17	3.80
(2,2)	1.71	1.71	1.49	2.35	2.35	1.85	2.91	2.91	2.38
(3,1)	2.02	2.13	1.95	2.68	2.81	2.47	3.36	3.53	3.15
(3,2)	1.49	1.65	1.62	1.89	2.20	2.06	2.41	2.76	2.63
(3,3)	2.03	2.16	1.98	2.53	2.67	2.48	3.25	3.44	3.18
(4,1)	1.95	1.95	1.82	2.66	2.66	2.41	3.30	3.30	3.02
(4,2)	1.43	1.58	1.60	1.80	2.10	2.02	2.30	2.63	2.58
(4,3)	2.10	2.08	1.83	2.63	2.61	2.30	3.37	3.34	2.94
(4,4)	2.06	2.05	2.07	2.66	2.62	2.69	3.36	3.33	3.39
	Chol-tr	**Chol-eig**	**Chol-det**	**Chol-tr**	**Chol-eig**	**Chol-det**	**Chol-tr**	**Chol-eig**	**Chol-det**
(1,1)	1.19	1.76	1.81	1.53	2.32	2.39	**1.94**	2.91	3.00
(2,1)	1.48	2.24	2.31	1.91	3.07	3.19	**2.42**	3.80	3.94
(2,2)	1.12	1.45	1.49	1.25	1.81	1.86	**1.68**	2.32	2.38
(3,1)	1.34	1.95	2.01	1.69	2.58	2.67	**2.15**	3.23	3.34
(3,2)	1.14	1.54	1.57	1.31	1.89	1.94	**1.74**	2.44	2.50
(3,3)	1.39	2.01	2.09	1.70	2.55	2.66	**2.20**	3.25	3.38
(4,1)	1.24	1.81	1.86	1.63	2.47	2.55	**2.05**	3.07	3.16
(4,2)	1.05	1.42	1.46	1.23	1.78	1.84	**1.62**	2.28	2.36
(4,3)	1.36	2.01	2.07	1.68	2.62	2.73	**2.16**	3.30	3.44
(4,4)	1.51	2.23	2.30	1.91	2.88	2.98	**2.44**	3.64	3.77

Abbreviations: EE, element-by-element; IM, importance mapping; rBias, relative bias; rRMSE, relative root mean square error; rSD, relative standard deviation.

results in the geometric average basket option, all the methods perform well in the estimation, and rBias, rSD, and rRMSE are small. Among all the methods, Chol-tr performs best in this example.

In addition, we know that the methods with Cholesky decomposition can guarantee the estimated matrix to be positive definite. While if we use EE and IM-tr, some of the estimated matrices may not be positive definite. So we define the average violation rate (aVR), that is, the percentage of the estimated matrices to be nonpositive definite, which is given by

$$aVR = 1 - \frac{1}{N_t R} \sum_{m=1}^{N_t} \sum_{r=1}^{R} \mathbb{1}\{\widehat{\mathbf{\Phi}}^r(\mathbf{x}_m^t) \text{ is positive definite}\}.$$

TABLE 4 Gamma estimation of arithmetic-mean call basket option with six underlying assets

	rBias (%)			rSD (%)			rRMSE (%)		
	EE	**IM-tr**	**Chol-EE**	**EE**	**IM-tr**	**Chol-EE**	**EE**	**IM-tr**	**Chol-EE**
(1,1)	4.56	3.70	4.47	1.46	1.99	2.02	4.90	4.31	5.02
(2,1)	5.05	3.82	4.21	1.62	2.07	1.81	5.41	4.46	4.73
(2,2)	4.23	4.02	4.18	2.49	2.25	2.26	5.07	4.73	4.90
(3,1)	3.99	3.19	3.46	1.75	1.72	1.37	4.47	3.72	3.84
(3,2)	4.44	3.32	3.31	1.41	1.80	1.65	4.77	3.88	3.81
(3,3)	3.96	2.78	2.77	0.51	1.49	1.47	4.02	3.25	3.24
(4,1)	4.57	3.60	4.05	1.54	1.99	1.69	4.87	4.21	4.51
(4,2)	5.04	3.70	3.90	1.37	2.07	2.01	5.35	4.35	4.50
(4,3)	4.51	3.16	3.24	0.45	1.79	1.62	4.55	3.74	3.74
(4,4)	4.26	3.56	3.80	1.85	2.00	1.91	4.73	4.19	4.37
	Chol-tr	**Chol-eig**	**Chol-det**	**Chol-tr**	**Chol-eig**	**Chol-det**	**Chol-tr**	**Chol-eig**	**Chol-det**
(1,1)	3.36	3.57	3.69	1.78	2.03	1.96	**3.92**	4.22	4.32
(2,1)	3.49	3.69	3.86	1.88	2.10	2.10	**4.10**	4.37	4.55
(2,2)	3.66	3.88	4.10	2.04	2.33	2.33	**4.36**	4.66	4.91
(3,1)	2.91	3.07	3.19	1.58	1.74	1.70	**3.42**	3.63	3.74
(3,2)	3.06	3.18	3.40	1.66	1.87	1.85	**3.60**	3.79	4.01
(3,3)	2.57	2.72	2.81	1.40	1.53	1.49	**3.03**	3.22	3.29
(4,1)	3.27	3.48	3.59	1.81	2.01	1.95	**3.85**	4.13	4.22
(4,2)	3.38	3.56	3.76	1.88	2.12	2.10	**4.01**	4.26	4.47
(4,3)	2.90	3.08	3.18	1.64	1.78	1.76	**3.45**	3.67	3.77
(4,4)	3.26	3.48	3.61	1.79	2.01	1.99	**3.86**	4.14	4.27

Abbreviations: EE, element-by-element; IM, importance mapping; rBias, relative bias; rRMSE, relative root mean square error; rSD, relative standard deviation.

TABLE 5 The average violation rate of positive semi-definiteness of EE and IM

		Two underlying assets	**Four underlying assets**	**Six underlying assets**
aVR (%)	EE	5.00	5.50	4.25
	IM	4.25	2.75	1.25

Abbreviations: EE, element-by-element; IM, importance mapping.

Then we can obtain Table 5, which tells us that the nonpositive definite cases for EE and IM-tr indeed happen. Although most of the cases are positive definite, we can still observe the violated results. Based on Table 5, IM outperforms EE from the viewpoint of maintaining positive definiteness.

Based on the numerical examples in this subsection, we conclude that the IM-based methods have a comparable estimation accuracy to EE. Moreover, the IM-based methods have a smaller computational burden than EE, so we recommend the IM-based methods in fitting matrix surface. Among the IM-based methods, IM-tr is more attractive than the other alternatives. If we consider the positive definiteness, Chol-tr is recommended especially for basket option with a large number of underlying assets. It has a low computational effort, and still maintains the positive definiteness and the desirable estimation accuracy of the gamma matrix.

4.4 | Delta-gamma approximation

The delta-gamma approximation is used to duplicate the derivative price if the underlying stock price changes via a quadratic function. The basic idea comes from the Taylor expansion that a function can be approximated through its first- and second-order sensitivities. The delta-gamma approximation is widely used in financial engineering research as well as in practice to hedge the risk of derivatives and portfolios corresponding to the fluctuation of underlying stocks or some other risk factors, see Glasserman, Heidelberger, and Shahabuddin (2000). For example, Glasserman et al. (2000) derives a variance reduction technique for VaR starting from the delta-gamma approximation. We consider a portfolio with multi-underlying assets, therefore, the gamma is a matrix.

Suppose that we have an arithmetic average basket call option whose price equals to $P(\mathbf{S}_{i-1})$ given stock price $\mathbf{S}_{i-1}$ at time t. The approximation of option price $P(\mathbf{S}_i)$ given $\mathbf{S}_i$ at time $t + \Delta t$ is approximated by

$$P(\mathbf{S}_i) = P(\mathbf{S}_{i-1}) + \Delta^\top \delta_s + \frac{1}{2}\delta_s^\top \Gamma \delta_s, \tag{12}$$

where $\delta_i = \mathbf{S}_i - \mathbf{S}_{i-1}$ is a column vector that calculates the change in each underlying stock price from time t to $t + \Delta t$, Δ (delta) and Γ (gamma) are estimated at $\mathbf{S}_{i-1}$, $i = 1, 2, \ldots$.

Obviously, a good accuracy of the estimated delta and gamma results in a good approximation. In this example, the arithmetic average basket call option is considered, and we want to investigate the performance of the delta-gamma approximation by using the estimated gamma via the proposed methods. For the delta, we use the IPA estimator with sample size 5×10^6 as the true delta. We consider the

TABLE 6 The accuracy of delta-gamma approximation of basket option via matrix kriging methods ($d = 4$)

	True	15.37	13.01	10.35	7.23	8.18	8.75	10.28	13.76	11.62	6.85	Average
Bias (10^{-2})	EE	1.26	2.88	**0.42**	1.74	0.04	1.27	1.04	1.41	0.95	4.37	1.54
	IM-tr	1.31	2.92	0.78	**1.74**	**0.00**	1.24	1.06	1.39	1.00	**4.21**	1.57
	Chol-EE	1.21	2.91	0.46	1.80	0.01	1.23	1.04	**1.31**	0.99	4.66	1.56
	Chol-tr	**1.14**	2.73	0.53	1.82	0.08	1.23	1.03	1.44	0.80	4.68	1.55
	Chol-eig	1.15	2.65	0.58	1.77	0.07	1.25	**1.02**	1.47	**0.76**	4.83	1.56
	Chol-det	1.16	**2.63**	0.49	1.76	0.07	**1.22**	1.04	1.38	0.78	4.82	**1.54**
SD (10^{-2})	EE	0.61	1.19	1.82	**0.30**	0.35	0.60	0.40	0.50	0.81	1.64	0.82
	IM-tr	**0.32**	1.15	1.31	0.31	**0.32**	0.19	**0.29**	0.50	0.85	1.68	0.69
	Chol-EE	0.36	1.11	1.25	0.37	0.37	0.24	0.38	**0.50**	0.91	1.61	0.71
	Chol-tr	0.34	**1.10**	1.22	0.38	0.39	0.21	0.35	0.60	**0.80**	**1.52**	**0.69**
	Chol-eig	0.34	1.19	**1.11**	0.41	0.37	**0.19**	0.36	0.65	0.88	1.64	0.72
	Chol-det	0.34	1.22	1.26	0.37	0.37	0.22	0.33	0.54	0.89	1.72	0.73
RMSE (10^{-2})	EE	1.40	3.11	1.87	**1.77**	0.36	1.41	1.12	1.50	1.25	4.67	1.84
	IM-tr	1.35	3.14	1.52	1.77	**0.32**	1.25	1.10	1.48	1.31	**4.53**	1.78
	Chol-EE	1.26	3.11	1.34	1.84	0.37	1.25	1.10	**1.40**	1.35	4.93	1.80
	Chol-tr	**1.19**	2.94	1.33	1.86	0.40	1.25	1.09	1.56	**1.13**	4.92	**1.77**
	Chol-eig	1.20	2.91	**1.26**	1.82	0.38	1.26	**1.08**	1.60	1.16	5.10	1.78
	Chol-det	1.21	**2.90**	1.35	1.80	0.38	**1.24**	1.09	1.48	1.18	5.12	1.78

Abbreviations: EE, element-by-element; IM, importance mapping; RMSE, root mean square error; SD, standard deviation.

basket option with four underlying assets, and the parameters are the same as previous experiments. Here we randomly choose 10 testing points $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_{10}$ by Latin hypercube sampling and we consider those points as the daily route of underlying price, that is, $\delta_i = \mathbf{S}_i - \mathbf{S}_{i-1}$, $i = 1, 2, \ldots, 10$. We fit the overall surface once and use the fitted gamma surface to track movements of option price. Here, one-step-ahead prediction is used, and we replicate the procedure $R = 20$ times.[3] Table 6 shows the approximation results in detail. According to bias, standard deviation and RMSE respectively, the top performance is marked as bold comparing among six methods: EE, IM-tr, EE-Chol, Chol-tr, Chol-eig, Chol-det. The first row shows the true option prices for 10 days (10 testing points), and the bias, SD, and RMSE of the differences between the estimated prices and the true prices for all the methods are calculated. The results indicate that all the methods work well in the delta-gamma approximation. Considering the computational burden and positive definiteness, Chol-tr is more attractive than other methods.

5 | CONCLUSION

In this paper, we propose several matrix kriging methods for fitting the gamma matrix surface, which is used in real-time risk management. We first derive the EE method, which treats the element in the matrix separately, and use stochastic kriging to obtain the surface of each element of the matrix. To overcome the challenge of high computational burden for large-dimensional matrices, the importance mapping method is proposed, and its theoretical properties are investigated. To maintain the positive definiteness of the matrices, the Cholesky decomposition method is proposed. Taking basket call options as examples, we study the effectiveness of the proposed methods. The importance mapping-based method turns out to have more desirable estimation results than other methods. In the future research, we may consider how to combine first- and second-order Greeks together to improve the accuracy of the estimation of both the first- and second-order Greeks. On the other hand, the efficient selection algorithm can be considered to enhance the performance of gamma estimation further.

[3]Here, one-step-ahead prediction means we use true value of $P(\mathbf{S}_{i-1})$ in (12) instead of estimated value.

ACKNOWLEDGMENTS

The research reported in this paper is partially supported by Hong Kong Research Grants Council Grant (GRF 16203214, 11504017), National Natural Science Foundation of China (Grants 71801148), and Shanghai Young Eastern Scholar Program (N.60-D129-18-202).

ORCID

L. Jeff Hong https://orcid.org/0000-0001-7011-4001
Guangxin Jiang https://orcid.org/0000-0002-2604-7750

REFERENCES

Ankenman, B., Nelson, B. L., & Staum, J. (2010). Stochastic kriging for simulation metamodeling. *Operations Research*, *58*, 371–382.

Björk, T. (2009). *Arbitrage theory in continuous time*. Oxford, England: Oxford University Press.

Broadie, M., & Glasserman, P. (1996). Estimating security price derivatives using simulation. *Management Science*, *42*, 269–285.

Chen, X., Ankenman, B., & Nelson, B. L. (2013). Enhancing stochastic kriging metamodels with gradient estimators. *Operations Research*, *61*, 512–528.

Cressie, N. A. C. (1993). *Statistics for spatial data*. New York, NY: Wiley.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap*. New York, NY: Chapman and Hall/CRC Press.

Elie, R., Fermanian, J. D., & Touzi, N. (2007). Kernel estimation of Greek weights by parameter randomization. *The Annals of Applied Probability*, *17*, 1399–1423.

Fu, M. C. (2006). Gradient estimation. *Handbooks in Operations Research and Management Science*, *13*, 575–616.

Fu, M. C., & Hu, J. Q. (1997). Conditional Monte Carlo: Gradient estimation and optimization applications. Norwell, MA: Kluwer Academic Publishers.

Glasserman, P. (2013). Monte Carlo methods in financial engineering. New York, NY: Springer Science & Business Media.

Glasserman, P., Heidelberger, P., & Shahabuddin, P. (2000). Variance reduction techniques for estimating value-at-risk. *Management Science*, *46*, 1349–1364.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, *33*(10), 75–84.

Heidergott, B., Vazquez-Abad, F., Pflug, G., & Farenhorst-Yuan, T. (2010). Gradient estimation for discrete-event systems by measure-valued differentiation. *ACM Transactions on Modeling and Computer Simulation*, *20*(1), 5.

Ho, Y. C., & Cao, X. (1983). Perturbation analysis and optimization of queueing networks. *Journal of Optimization Theory and Applications*, *40*, 559–582.

Hong, L. J., & Jiang, G. (2018). *Offline simulation online application: A new framework of simulation-based decision making* (Under review).

Jiang, G., Hong, L. J., & Nelson, B. L. (2019). Online risk monitoring using offline simulation. *INFORMS Journal on Computing* (in press).

Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, *192*, 707–716.

Liu, G., & Hong, L. J. (2011). Kernel estimation of the greeks for options with discontinuous payoffs. *Operations Research*, *59*, 96–108.

Meyer, C. D. (2013). Matrix analysis and applied linear algebra. Philadelphia, PA: Cambridge University Press.

Peng, Y. J., Fu, M. C., Hu, J. Q., & Heidergott, B. (2018). A new unbiased stochastic derivative estimator for discontinuous sample performances with structural parameters. *Operations Research*, *66*(2), 487–499.

Pflug, G. C. (1989). Sampling derivatives of probabilities. *Computing*, *42*, 315–328.

Reiman, M. I., & Weiss, A. (1989). Sensitivity analysis for simulations via likelihood ratios. *Operations Research*, *37*(5), 830–844.

Schoutens, W. (2003). Lévy processes in finance. Chichester, England: Wiley.

Stein, M. L. (1999). Interpolation of spatial data: Some theory for kriging. New York, NY: Springer.

van Beers, W. C. M., & Kleijnen, J. P. C. (2003). Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, *54*, 255–262.

Wang, Y., Fu, M. C., & Marcus, S. I. (2012). A new stochastic derivative estimator for discontinuous payoff functions with application to financial derivatives. *Operations Research*, *60*(2), 447–460.

How to cite this article: Yun X, Hong LJ, Jiang G, Wang S. On gamma estimation via matrix kriging. *Naval Research Logistics* 2019;66:393–410. https://doi.org/10.1002/nav.21850

APPENDIX A: PROPERTY VIOLATION OF EE: AN EXAMPLE

EE method is sometimes accompanied with violation of positive definiteness for gamma estimation even by simple kriging (Cressie, 1993). A specific example is shown. Consider an arithmetic average call basket option with two underlying asset S_1 and S_2. We set strike price $K = 100$, risk-free interest rate $r = 3\%$, and expiry date $T = 1$. The discounted expected payoff function of this basket option is set as

$$P = e^{-rT}\mathbb{E}\left[\left(\frac{1}{d}\sum_{i=1}^{d} S_i(T) - K\right)^+\right].$$

By Latin hypercube sampling, we randomly choose six design points

$$\mathbf{x}_1 = [107.49; 102.27], \mathbf{x}_2 = [104.43; 98.78],$$
$$\mathbf{x}_3 = [105.23; 97.10], \mathbf{x}_4 = [101.04; 100.19],$$
$$\mathbf{x}_5 = [102.31; 95.21], \mathbf{x}_6 = [108.56; 104.78],$$

and estimate the response matrix at each design point $n = 150$ times. By calculation the sample mean of estimated gamma are all positive-definite, with positive minimum eigenvalue $\lambda_1^{\min} = 4.06 \times 10^{-4}$, $\lambda_2^{\min} = 6.69 \times 10^{-4}$, $\lambda_3^{\min} = 3.75 \times 10^{-4}$, $\lambda_4^{\min} = 4.41 \times 10^{-4}$, $\lambda_5^{\min} = 4.34 \times 10^{-4}$, $\lambda_6^{\min} = 4.83 \times 10^{-4}$. As a special case, we let testing point $\mathbf{x}_0 = [110.25;100]$ and apply ordinary kriging to six design points above to predict the response gamma matrix at $\mathbf{x}_0$. It turns out that the prediction

$$\widehat{\mathbf{\Phi}}(\mathbf{x}_0) = \begin{pmatrix} 3.56 \times 10^{-3} & 3.96 \times 10^{-3} \\ 3.96 \times 10^{-3} & 3.16 \times 10^{-3} \end{pmatrix}$$

is not positive with minimum eigenvalue $\lambda_0^{\min} = -5.88 \times 10^{-4}$. However, one can prove that the gamma matrix should be positive semi-definite for convex payoff mathematically.

APPENDIX B: PROOF OF THEOREM 1

Proof Recall that $\overline{\mathbf{\Pi}} = (\overline{\Pi}(\mathbf{x}_1), \overline{\Pi}(\mathbf{x}_2), \ldots, \overline{\Pi}(\mathbf{x}_N))$, and

$$\overline{\Pi}(\mathbf{x}_i) = \beta_{I0} + \mathsf{M}_I(\mathbf{x}_i) + \overline{\varepsilon}_I(\mathbf{x}_i),$$

where $\overline{\varepsilon}_I(\mathbf{x}_i) = 1/n_i \sum_{h=1}^{n_i} \varepsilon_I^h(\mathbf{x}_i)$. Since $\varepsilon_I^h(\mathbf{x}_i)$ is normal distribution, so the estimated variance given by (8) is independent of $\overline{\varepsilon}_I(\mathbf{x}_i)$. So

$$\mathbb{E}[\hat{\hat{\pi}}(\mathbf{x}_0)] = \mathbb{E}\left[\mathbb{E}\left[\left(\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot) + \mathbf{1}_N \frac{1-\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot)}{\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\overline{\boldsymbol{\Pi}}\,\middle|\,\hat{\boldsymbol{\Sigma}}_{\epsilon_I}\right]\right]$$

$$= \mathbb{E}\left[\left(\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot) + \mathbf{1}_N \frac{1-\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot)}{\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N\beta_{I0}\right]$$

$$= \beta_{I0}\,\mathbb{E}[\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot)^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N + 1 - \mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot)]$$

$$= \beta_{I0}.$$

Therefore $\mathbb{E}[\hat{\hat{\pi}}(\mathbf{x}_0) - \pi(\mathbf{x}_0)] = \beta_{I0} - \beta_{I0} = 0$. ■

APPENDIX C: PROOF OF THEOREM 2

Proof According to the proof of Theorem 1,

$$\hat{\mathbf{w}}^*(\mathbf{x}_0)^\top\mathbf{1}_N = \left(\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot) + \mathbf{1}_N \frac{1-\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\boldsymbol{\Sigma}_{\mathsf{M}_I}(\mathbf{x}_0,\cdot)}{\mathbf{1}_N^\top(\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N}\right)^\top (\boldsymbol{\Sigma}_{\mathsf{M}_I}+\hat{\boldsymbol{\Sigma}}_{\epsilon_I})^{-1}\mathbf{1}_N = 1. \tag{C1}$$

Ψ is a linear mapping such that $\pi(\mathbf{x}) = \sum_{i=1}^d\sum_{j=1}^d u_{i,j}\Phi_{i,j}(\mathbf{x})$. According to Assumption 2, sample variance at each design point given by

$$\hat{V}(\mathbf{x}_k) = \frac{1}{n_k-1}\sum_{h=1}^{n_k}\left(\sum_{i=1}^d\sum_{j=1}^d u_{i,j}\epsilon_{i,j}^h(\mathbf{x}_k) - \sum_{i=1}^d\sum_{j=1}^d u_{i,j}\bar{\epsilon}_{i,j}(\mathbf{x}_k)\right)^2 \tag{C2}$$

is independent of the each coordinate in $\overline{\mathbf{H}}_{i,j}$ for all i and j, then

$$\begin{aligned}\mathbb{E}[\hat{\Phi}_{i,j}(\mathbf{x}_0)] &= \mathbb{E}[\hat{\mathbf{w}}^*(\mathbf{x}_0)^\top\overline{\mathbf{H}}_{i,j}]\\ &= \mathbb{E}[\hat{\mathbf{w}}^*(\mathbf{x}_0)^\top\,\mathbb{E}[\overline{\mathbf{H}}_{i,j}|\hat{\boldsymbol{\Sigma}}_{\epsilon_I}]]\\ &= \beta_{i,j}\mathbb{E}[\hat{\mathbf{w}}^*(\mathbf{x}_0)^\top\mathbf{1}] = \beta_{i,j}.\end{aligned}$$

Therefore, $\mathbb{E}[\hat{\Phi}_{i,j}(\mathbf{x}_0) - \Phi_{i,j}(\mathbf{x}_0)] = 0$. ■

APPENDIX D: SPA ESTIMATORS REALIZATION OF GAMMA OF ARITHMETIC AVERAGE BASKET CALL OPTION

We consider an arithmetic average basket call option with two underlying stocks following geometric Brownian motion, whose initial value $\mathbf{S}_0 = [105, 100]^\top$, volatility $\boldsymbol{\sigma} = [0.3, 0.4]^\top$, strike price $K = 100$ and expiry time $T = 1$. Here, it is to show that the gamma matrix of arithmetic average basket call option, whose elements are calculated by the SPA estimator in (F2), cannot maintain positive definite for all generated sample path of stock price at time T, denoted by $\mathbf{S}(T) = (S_1(T), S_2(T))^\top$. For example, if we let $\mathbf{S}(T) = (113, 56)^\top$, then the realization of the SPA estimator is

$$\hat{\Gamma} = \begin{pmatrix} 4.58\times10^{-3} & 5.08\times10^{-3}\\ 5.08\times10^{-3} & 4.11\times10^{-3}\end{pmatrix},$$

and the minimum eigenvalue is -7.38×10^{-4}. If we let $\mathbf{S}(T) = (90, 109)^\top$, the realization of the SPA estimator is

$$\hat{\Gamma} = \begin{pmatrix} 4.86\times10^{-3} & 3.88\times10^{-3}\\ 3.88\times10^{-3} & 4.98\times10^{-3}\end{pmatrix},$$

and the minimum eigenvalue is 8.81×10^{-3}.

APPENDIX E: SPA ESTIMATOR OF GAMMA OF GEOMETRIC AVERAGE BASKET CALL OPTION

The expected payoff function of basket option is

$$c^G = e^{-rT}\mathbb{E}\left[\left(\sqrt[d]{S_1(T)S_2(T)\,\ldots\,S_d(T)} - K\right)\mathbb{1}_{\left\{\sqrt[d]{\prod_{k=1}^d S_k(T)}>K\right\}}\right],$$

Let

$$Y^G = \left(\sqrt[d]{S_1(T)S_2(T)\,\ldots\,S_d(T)} - K\right)\mathbb{1}_{\left\{\sqrt[d]{\prod_{k=1}^d S_k(T)}>K\right\}}.$$

Note that Y^G is Lipchitz continuous, thus we can interchange the expectation and derivative, that is

$$\begin{aligned}\partial\mathbb{E}(Y^G)/\partial S_i(0) &= \mathbb{E}[\partial Y^G/\partial S_i(0)]\\ &= \mathbb{E}\left[\frac{\sqrt[d]{S_1(T)S_2(T)\,\ldots\,S_d(T)}}{dS_i(0)}\mathbb{1}_{\left\{\sqrt[d]{S_1(T)S_2(T)\cdots S_d(T)}>K\right\}}\right]\\ &= \mathbb{E}\left[\mathbb{E}\left(\frac{\sqrt[d]{S_1(T)S_2(T)\,\ldots\,S_d(T)}}{dS_i(0)}\mathbb{1}_{\left\{\sqrt[d]{\prod_{k=1}^d S_k(T)}>K\right\}}|S_k(T), k\neq i\right)\right]\end{aligned}$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{d} E\left(\frac{(S_i(T))^{1/d}}{S_i(0)} \mathbb{1}_{\left\{S_i(T)>K^d/\prod_{k\neq i}^{d} S_k(T)\right\}} | S_k(T), k\neq i\right)\right]$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{n} \int_{K/\prod_{k\neq i}^{d} S_k(T)}^{\infty} \frac{(S_i(T))^{1/n}}{S_i(0)} dF(S_i(T))\right]$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{dS_i(0)} \left(E[(S_i(T))^{1/d}] - \int_0^{\frac{K^d}{\prod_{k\neq i}^{d} S_k(T)}} (S_i(T))^{1/d} dF(S_i(T))\right)\right]$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{dS_i(0)} \left((S_i(0))^{1/d} e^{\frac{(r-1/2\sigma_i^2)T}{d} + \frac{\sigma_i^2 T}{2d^2}} - \int_0^{\frac{K^d}{\prod_{k\neq i}^{d} S_k(T)}} (S_i(T))^{1/d} dF(S_i(T))\right)\right].$$

Let

$$\Pi = \frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{dS_i(0)} \left((S_i(0))^{1/d} e^{\frac{(r-1/2\sigma_i^2)T}{d} + \frac{\sigma_i^2 T}{2d^2}} - \int_0^{\frac{K^d}{\prod_{k\neq i}^{d} S_k(T)}} (S_i(T))^{1/d} dF(S_i(T))\right).$$

Therefore, the SPA estimator of the diagonal element in the gamma matrix is

$$\hat{\Gamma}^G(i,i) = \frac{\partial \Pi}{\partial S_i(0)}$$

$$= \frac{1}{d^2 (Si(0))^2} \sqrt[d]{\prod_{k\neq i}^{n} S_k(T)} e^{\frac{\mu^G}{d} + \frac{(\sigma^G)^2}{2d^2}} (1-d)(S_i(0))^{1/d}$$

$$+ \frac{\sqrt[d]{\prod_{k\neq i}^{d} S_k(T)}}{d(S_i(0))^2} \int_0^{K^n/\prod_{k\neq i}^{n} S_k(T)} \frac{x^{1/d-1}}{\sigma^G}$$

$$\phi\left(\frac{\log(x/(S_i(0))) - \mu^G}{\sigma^G}\right)\left[1 - \frac{\log(x/S_i(0)) - \mu^G}{(\sigma^G)^2}\right] dx,$$

where $\mu^G = (r - 1/2\sigma_i^2)T$ and $\sigma^G = \sigma_i^2 T$. Similarly, for any $j \neq i$, we have

$$\frac{\partial \mathbb{E}(Y^G)}{\partial S_i(0)} = \mathbb{E}\left[\frac{\partial Y}{\partial S_i(0)}\right]$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{S_1(T)S_2(T)\cdots S_d(T)}}{dS_i(0)} \mathbb{1}_{\left\{\sqrt[d]{\prod_{k=1}^{d} S_k(T)} > K\right\}}\right]$$

$$= \mathbb{E}\left[\frac{\sqrt[d]{\prod_{k\neq j}^{d} S_k(T)}}{dS_i(0)} \left(\mathbb{E}[(S_j(T))^{1/d}] - \int_0^{\frac{K^d}{\prod_{k\neq j}^{d} S_k(T)}} (S_j(T))^{1/d} dF(S_j(T))\right)\right].$$

Let

$$\Pi = \frac{\sqrt[d]{\prod_{k\neq j}^{d} S_k(T)}}{dS_i(0)} \left((S_j(0))^{1/d} e^{\frac{(r-1/2\sigma_j^2)T}{n} + \frac{\sigma_j^2 T}{2n^2}} - \int_0^{\frac{K^n}{\prod_{k\neq j}^{n} S_k(T)}} (S_j(T))^{1/d} dF(S_j(T))\right).$$

Then, if we take derivative with respect to $S_j(0)$, we have the SPA estimator for nondiagonal elements,

$$\hat{\Gamma}^G(i,j) = \frac{\partial \Pi}{\partial S_j(0)} = \frac{\sqrt[d]{\prod_{k\neq j}^{d} S_k(T)}}{dS_i(0)S_j(0)} \left[e^{\frac{\mu^G}{d} + \frac{(\sigma^G)^2}{2d^2}} (S_j(0))^{1/d} - \int_0^{\frac{K^d}{\prod_{k\neq j}^{d} S_k(T)}} \frac{x^{1/d-1}}{(\sigma^G)^3} \phi\left(\frac{\log(x/(S_j(0))) - \mu^G}{\sigma^G}\right) (\log(x/S_j(0)) - \mu^G) dx\right].$$

APPENDIX F: SPA ESTIMATOR OF GAMMA OF ARITHMETIC AVERAGE BASKET CALL OPTION

Let the payoff of basket option be

$$Y^A = \left(\frac{1}{d}\sum_{i=1}^{d} S_i(T) - K\right) \mathbb{1}_{\left\{\frac{1}{d}\sum_{i=1}^{d} S_i(T) > K\right\}}.$$

Note that Y^A is Lipchitz continuous.

$$\frac{\partial \mathbb{E}(Y^A)}{\partial S_i(0)} = \mathbb{E}\left[\frac{\partial Y^A}{\partial S_i(0)}\right] = \mathbb{E}\left[\frac{S_i(T)}{dS_i(0)} \mathbb{1}_{\left\{\frac{1}{d}\sum_{i=1}^{d} S_i(T) > K\right\}}\right]$$

$$= \mathbb{E}\left[\mathbb{E}\left(\frac{S_i(T)}{dS_i(0)} \mathbb{1}_{\left\{\frac{1}{d}\sum_{i=1}^{d} S_i(T) > K\right\}} | S_k(T), k\neq i\right)\right]$$

$$= \mathbb{E}\left[\frac{1}{dS_i(0)} \int_{dK - \sum_{k\neq i} S_k(T)}^{\infty} S_i(T) dF(S_i(T))\right]$$

by conditioning on $S_k(T)$ for $i \neq k$. Here, $F(S_i(T))$ denotes the cumulative distribution function of $S_i(T)$, which is a normal distribution. If $dK \geq \sum_{k\neq i} S_k(T)$, then

$$\begin{aligned}\frac{\partial^2 \mathbb{E}(Y^A)}{\partial S_i(0)^2} &= \frac{\partial}{\partial S_i(0)} \mathbb{E}\left[\frac{1}{dS_i(0)} \int_{dK-\sum_{k\neq i} S_k(T)}^{\infty} S_i(T) dF(S_i(T))\right] \\ &= \mathbb{E}\left[\frac{\partial}{\partial S_i(0)}\left(\frac{1}{dS_i(0)} \int_{dK-\sum_{k\neq i} S_k(T)}^{\infty} S_i(T) dF(S_i(T))\right)\right] \\ &= \frac{1}{n(S_i(0))^2 \sigma_i \sqrt{T}} \int_0^{nK-\sum_{k\neq i} S_k(T)} \phi\left(\frac{\rho(S_i(0))}{\sigma^A}\right) \\ &\quad \times \left[1 - \frac{\rho(S_i(0))}{(\sigma_i^A)^2}\right] dx,\end{aligned}$$

where $\phi(\cdot)$ is the standard normal density function and

$$\rho(S_i(0)) = \log(x/S_i(0)) - \mu_i^A T,$$

$\mu_i^A = r - 1/2\sigma_i^2$, and $\sigma_i^A = \sigma_i \sqrt{T}$, $i = 1, 2, \ldots, d$.
If $dK < \sum_{k\neq i} S_k(T)$, then

$$\frac{\partial \mathbb{E}(Y)}{\partial S_i(0)} = \frac{e^{rT}}{d}.$$

So the second order gradient is zero. The SPA estimator for diagonal element is

$$\widehat{\Gamma}(i,i) = \begin{cases} \frac{1}{d(S_i(0))^2 \sigma_i^A} \int_0^{U_i} \phi\left(\frac{\rho(S_i(0))}{\sigma_i^A}\right)\left[1 - \frac{\rho(S_i(0))}{(\sigma_i^A)^2}\right] dx, \\ \qquad dK \geq \sum_{k\neq i} S_k(T) \\ 0, \quad dK < \sum_{k\neq i} S_k(T) \end{cases} \tag{F1}$$

where

$$U_i = dK - \sum_{k\neq i} S_k(T).$$

For any $j \neq i$, we have

$$\begin{aligned}\frac{\partial \mathbb{E}(Y^A)}{\partial S_i(0)} &= \mathbb{E}\left[\frac{\partial Y^A}{\partial S_i(0)}\right] = \mathbb{E}\left[\frac{S_i(T)}{dS_i(0)} 1_{\left\{\frac{1}{d}\sum_{i=1}^d S_i(T) > K\right\}}\right] \\ &= \mathbb{E}\left[\mathbb{E}\left(\frac{S_i(T)}{dS_i(0)} 1_{\left\{\frac{1}{d}\sum_{i=1}^d S_i(T) > K\right\}} | S_k(T), k \neq j\right)\right] \\ &= \mathbb{E}\left[\frac{S_i(T)}{dS_i(0)} \int_{U_j}^{\infty} dF(S_j(T))\right].\end{aligned}$$

If $dK \geq \sum_{k\neq j} S_k(T)$, then

$$\begin{aligned}\frac{\partial \mathbb{E}(Y^A)}{\partial S_i(0)} &= \mathbb{E}\left[\frac{S_i(T)}{dS_i(0)}\left(1 - \int_0^{U_j} dF(S_j(T))\right)\right] \\ &= \mathbb{E}\left[\frac{S_i(T)}{nS_i(0)} - \frac{S_i(T)}{nS_i(0)}\right. \\ &\quad \left.\Phi\left(\frac{\log(U_j) - \log(S_j(0)) - (r - 1/2\sigma_j^2)T}{\sigma_j \sqrt{T}}\right)\right].\end{aligned}$$

Take derivative with respect to $S_j(0)$,

$$\begin{aligned}\frac{\partial^2 \mathbb{E}(Y^A)}{\partial S_i(0) \partial S_j(0)} &= \mathbb{E}\left[\frac{\partial}{\partial S_j(0)}\left(\frac{S_i(T)}{nS_i(0)} - \frac{S_i(T)}{nS_i(0)}\right.\right. \\ &\quad \left.\left.\Phi\left(\frac{\log(U_j) - \log(S_j(0)) - (r - 1/2\sigma_j^2)T}{\sigma_j \sqrt{T}}\right)\right)\right] \\ &= \mathbb{E}\left[\frac{S_i(T)}{d\sigma_j \sqrt{T} S_i(0) S_j(0)}\right. \\ &\quad \left.\phi\left(\frac{\log(U_j) - \log(S_i(0)) - (r - 1/2\sigma_j^2)T}{\sigma_j \sqrt{T}}\right)\right].\end{aligned}$$

If $dK \geq \sum_{k\neq j} S_k(T)$, then

$$\frac{\partial \mathbb{E}(Y)}{\partial S_i(0)} = \frac{S_i(T)}{dS_i(0)}.$$

The SPA estimator is

$$\widehat{\Gamma}(i,j) = \begin{cases} \frac{S_i(T)}{d\sigma_j^A S_i(0) S_j(0)} \phi\left(\frac{\log(U_j) - \log(S_i(0)) - \mu_j^A T}{\sigma_j^A}\right), \\ \qquad dK \geq \sum_{k\neq i} S_k(T) \\ 0, \quad dK < \sum_{k\neq i} S_k(T) \end{cases}. \tag{F2}$$

APPENDIX G: ANALYTICAL FORMULA OF THE GAMMA OF GEOMETRIC AVERAGE BASKET CALL OPTION

According to settings in numerical experiment, there are underlying stocks for the geometric-mean basket call option. And we assume that each underlying stock follows $S_i \sim GBM(\mu, \sigma_i^2)$. Let $\widetilde{S}_T = \sqrt[n]{S_1(T) S_2(T) \ldots S_n(T)}$, we have

$$\begin{aligned}\log(\widetilde{S}_T) &= \frac{1}{d}\sum_{i=1}^n \log(S_i(T)) \\ &= \frac{1}{d}\sum_{i=1}^d \left[\log(S_i(0) + \left(\mu - \frac{1}{2}\sigma_i^2\right)T + \sigma_i B_i(T)\right] \\ &= \frac{1}{d}\sum_{i=1}^d \log(S_i(0)) + \frac{1}{d}\sum_{i=1}^d \left(\mu - \frac{1}{2}\sigma_i^2\right)T \\ &\quad + \frac{1}{d}\sum_{i=1}^d \sigma_i B_i(T).\end{aligned}$$

For any $i, j \in \{1, 2, \ldots, d\}$, $i \neq j$, we have $Cov(B_i(t), B_j(t)) = 0$. Accordingly,

$$Var\left[\sum_{i=1}^d \sigma_i B_i(T)\right] = \sum_{i=1}^d Var[\sigma_i B_i(T)] = \sum_{i=1}^d \sigma_i^2 T.$$

So, fix T,

$$\frac{1}{d}\sum_{i=1}^d \sigma_i B_i(T) \sim \Phi\left(0, \frac{1}{d^2}\sum_{i=1}^d \sigma_i^2 T\right).$$

Then we have

$$\widetilde{S}_T = \sqrt[d]{S_1(0)S_2(0)\ \dots\ S_d(0)}e^{\left(\mu-\frac{d}{2}\widetilde{\sigma}^2\right)T+\widetilde{\sigma}\sqrt{T}Z},$$

where

$$\widetilde{\sigma}^2 = \frac{1}{d^2}\sum_{i=1}^{d}\sigma_i^2.$$

Therefore, $\widetilde{S}_T$ has the distribution $GBM\left(\left[r-\left(\frac{d}{2}\widetilde{\sigma}^2-\frac{1}{2}\widetilde{\sigma}^2\right)\right],\widetilde{\sigma}^2\right)$ under the risk-neutral framework. And the value at time zero is

$$\widetilde{S}_0 = e^{-rT}E[\widetilde{S}_T] = \sqrt[d]{S_1(0)S_2(0)\ \dots\ S_d(0)}e^{-(\frac{d}{2}\widetilde{\sigma}^2T-\frac{1}{2}\widetilde{\sigma}^2T)}.$$

Its European call option is given by the Black-Scholes formula,

$$c^G = \widetilde{S}_0\Phi(d_1) - Ke^{-rT}\Phi(d_2),$$

where

$$d_{1,2} = \frac{\log\widetilde{S}_0/K + \left(r\pm\frac{1}{2}\widetilde{\sigma}^2\right)T}{\widetilde{\sigma}\sqrt{T}},$$

that is, the pricing formula for geometric-mean basket option is as follow:

$$c^G = e^{-\frac{d-1}{2}\widetilde{\sigma}^2T}\sqrt[d]{S_1(0)\ \dots\ S_d(0)}\,\Phi\left(\frac{\log\left(\sqrt[d]{S_1(0)\ \dots\ S_d(0)}/K\right)+\left(r-\frac{d-2}{2}\widetilde{\sigma}^2\right)T}{\widetilde{\sigma}\sqrt{T}}\right) - Ke^{-rT}\Phi\left(\frac{\log\left(\sqrt[d]{S_1(0)\ \dots\ S_d(0)}/K\right)+\left(r-\frac{d}{2}\widetilde{\sigma}^2\right)T}{\widetilde{\sigma}\sqrt{T}}\right).$$

Specifically, let $n = 2$, the basket option is then a spread option and the pricing formula is as follow:

$$c^G = e^{-\frac{1}{2}\widetilde{\sigma}^2T}\sqrt{S_1(0)S_2(0)}\Phi\left(\frac{\log(\sqrt{S_1(0)S_2(0)}/K)+rT}{\widetilde{\sigma}\sqrt{T}}\right) - Ke^{-rT}\Phi\left(\frac{\log(\sqrt{S_1(0)S_2(0)}/K)+(r-\widetilde{\sigma}^2)T}{\widetilde{\sigma}\sqrt{T}}\right),$$

where

$$\widetilde{\sigma}^2 = \frac{1}{4}(\sigma_1^2+\sigma_2^2),$$

that is, the pricing formula for geometric-mean basket option is as follow:

$$\frac{\partial c^G}{\partial S_i(0)} = \frac{\sqrt[d]{S_1(0)\ \dots\ S_d(0)}e^{-\frac{d-1}{2}\widetilde{\sigma}^2T}}{dS_i(0)}\left[\Phi(d_1)+\frac{\phi(d_1)}{\widetilde{\sigma}\sqrt{T}}\right] - \frac{e^{-rT}K}{d\widetilde{\sigma}\sqrt{T}S_i(0)}\phi(d_2).$$

Then it follows the second-order derivative, for any $i = 1, 2, \dots, n$ and any $j \neq i$.

$$\frac{\partial^2 c^G}{\partial S_i(0)^2} = \frac{e^{-\frac{d-1}{2}\widetilde{\sigma}^2T}\sqrt[d]{\prod_{j=1}^{d}S_j(0)}}{(dS_i(0))^2}\times\left[(1-d)\Phi(d_1)+\frac{2-d}{\widetilde{\sigma}\sqrt{T}}\phi(d_1)-\frac{d_1}{\widetilde{\sigma}^2T}\phi(d_1)\right] + \frac{e^{-rT}K\phi(d_2)}{n\widetilde{\sigma}\sqrt{T}(S_i(0))^2}\left(1+\frac{d_2}{d\widetilde{\sigma}\sqrt{T}}\right)$$

and

$$\frac{\partial^2 c^G}{\partial S_i(0)\partial S_j(0)} = \frac{e^{-\frac{d-1}{2}\widetilde{\sigma}^2T}\sqrt[d]{\prod_{k=1}^{d}S_k(0)}}{d^2S_i(0)S_j(0)}\times\left[\Phi(d_1)+\frac{\phi(d_1)}{\widetilde{\sigma}\sqrt{T}}\left(2-\frac{d_1}{\widetilde{\sigma}\sqrt{T}}\right)\right] + \frac{e^{-rT}Kd_2}{n^2\widetilde{\sigma}^2TS_i(0)S_j(0)}\phi(d_2).$$

They correspond to the diagonal elements and nondiagonal elements of gamma Γ^G. Basically, we apply the above expression for calculating true value of gamma we want to estimate via proposed methods.