Application of outer approximation to forecasting losses and scenarios in the target of portfolios with high of nonlinear risk

Katsuhiro Tanaka

Abstract

The purpose of this paper is to find appropriate solutions to concave quadratic programming using outer approximation algorithm, which is one of the algorithm of global optimization, in the target of the strong of concavity of object function i.e. high of nonlinear risk of portfolio. Firstly, my target model is a mathematical optimization programming to forecast scenarios and losses in the case of realistic interest rate portfolio. Object function of the model is quadratic function for representing portfolio loss using delta, gamma and vega. And, it becomes concave quadratic programming i.e. nonlinear programming, and is not guaranteed to find appropriate solution in using some general solver. However, it is possible to find appropriate solution in using a specific outer approximation algorithm. Secondly, my target case is in the high of nonlinear risk of portfolio, because main business of financial institution is to sell various financial product such as options, and then they are obliged to have highly nonlinear risk. Major nonlinear risk is vega, which tend to be negative in selling options. And negative of vega make quadratic function concave, because diagonal component of quadratic coefficient matrix become negative. Therefore, my target case is in the strong of concavity of object function. Lastly, I check that there exists relationship of negative correlation between high of nonlinear risk i.e. strong of concavity of quadratic function and calculation cost. Therefore, when it apples outer approximation algorithm to my target model and case, I see that it is very useful from the perspective of computation as well as risk management.

I believe that this paper will be of interest to researchers and practitioners in the field of market risk management in the financial industry.

Keyword: Risk management, Mathematical optimization, Outer approximation, Concave quadratic objective function, Forecasting of scenario and loss.

1 Introduction

In [15], the author proposed a mathematical programming for forecasting scenarios and losses in the target of realistic interest rate portfolios. In this paper, target is same. The point are that quadratic function in object function become CONCAVE from the perspective of risk management and mathematical optimization programming.

In risk management, actual financial institution have a portfolio with higher of nonlinear risk basically. It is normal for invest bank and security company, which sell various financial products such as options including a lot of risk called delta, gamma and vega. When I focus vega as major nonlinear risk, selling option means to be short vega i.e. negative of vega. In this case, invest bank and security company work vega hedge operation, because increasing volatility leads to increase loss,. However, products to do it such as vanilla option are not rich in financial market, in particular ITM and OTM. So, their portfolios always exist high non linear risk, portfolio managers of them are worrying how to manage high of nonlinear risk they can't help having in business.

In mathematical optimization programming, similar to [15], I can represent portfolio loss using quadratic function with delta, gamma, and vega. In this case, short vega leads diagonal component of quadratic coefficient matrix to be negative, and then negative of quadratic coefficient matrix i.e. negative of vega leads quadratic coefficient matrix to be non positive definite. As a result, quadratic function become concave inevitability.

Details of target programming of [15] were that object function was a concave quadratic function and constraint condition was nonlinear equality. So it classified nonlinear programming which isn't easy to find global solution. So, [15] found usable solutions by introducing original heuristic approach in order to prevent the sign of the solution from being inverse intentionally.

However, this approach is not expected to deliver appropriate solutions, in particular in the case of high of nonlinear risk of portfolio i.e. strong of concavity of the object function. Thus, [15] raised one of the future tasks that is to find more optimal solutions.

Regarding this task, [17] suggested an approach to find the global solution of a concave quadratic object function. In summarize of this approach, it was reformulated to a bilinear function and used an outer approximation to create a convex polyhedral set by adding cutting plane repeatedly, and looked for the best solutions, based on [10]. The authors demonstrated that solutions by this approach were almost the same as global solutions by an exact algorithm. In addition,they concluded that this approach was able to solve within a practical amount of time at up to rank five.

Here, there are three differences between my target model and [17], as follows.

Table 1: Differences between my target model and [17]

	My target model	[17]
A.Resource of coefficient of bilinear function	Realistic interest rate portfolio	Uniform random variable
B.Constraint condition	Nonlinear equality and linear inequality	Linear inequality
C.At most rank	Fourteen	Five

All of them don't become decisive points in finding appropriate solutions using outer approximation algorithm in the theory. However, in terms of A, I need to investigate whether there is something trend or not in focusing more realistic case for risk management in analysis, because concavity of object function become stronger. Also, in terms of other, I need to check whether it can solve within a practical amount of calculation cost or not in the computation, because the programming is more complicated.

The next section describes i) the dynamic conditional correlation-generalized autoregressive conditional heteroskedasticity (DCC-GARCH) model framework for representing movement of risk factor, volatility and correlation and ii) the formulation of mathematical programming for representing losses in the case of realistic interest rate portfolios and uncertainty of all risk factors. Additionally, the risk of portfolio is represented by losses of each greek part (delta, vega and gamma) using a normal model. Section 3 is devoted to an algorithm for finding appropriate solutions to forecast scenarios and losses using an outer approximation according to [17]. Computational results and analysis are presented in Section 4.

I show that outer approximation algorithm is very useful in the case of portfolio with high of nonlinear risk, because there exist relationship of negative correlation between the degree of concavity of quadratic function in the objective function and size of calculation cost. In addition, the algorithm can find appropriate solutions within a practical amount of calculation cost and I suggest suitable level of each parameter for calculation.

2 Formulation

2.1 Notation

Let $M(m \in \{1,\ldots,M\})$ be the grid number of a market rate, $\boldsymbol{T} \in \mathbf{R}^M$ be the vectors of the spot grid, and $I(i \in \{1,\ldots,I\})$ be the trade count. Then, let $T_s^{(i)}$ be the option term of trade i, $T_e^{(i)}$ be the expiry of the option, and $T_{e-s}^{(i)}$ be defined as $T_e^{(i)} - T_s^{(i)} (s, e \in \{1,\ldots,M\}$ and $s < e)$.

Let $t = 0$ be the base date, $t \in \{1,\ldots,T\}$ be the future period, $r_{T_m,t}$ be the market rate of T_m year at t, and $f_t^{(i)}$ be the forward swap rate of underlying of trade i.

Let $\sigma_{T_m,t}^S$ be the spot normal volatility of T_m year, and let $\sigma_{f_t^{(i)}}$ be the forward normal volatility of trade i.

Let $\boldsymbol{dr}_t \in \mathbf{R}^M$ be a vector of $r_{T_m,t}$, $\boldsymbol{\mu}_t \in \mathbf{R}^M$ be the drift term of an RF, and $\boldsymbol{\sigma}_t \in \mathbf{R}^M$ be a vector of $\sigma_{T_m,t}^S$.

Let $\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbf{R}^M$ denote a vector of the model parameters for representing volatility, and let a, b be the model parameters for representing correlation.

Let $\boldsymbol{Q}_t \in \mathbf{R}^{M\times M}$ be the variance-covariance matrix, $\boldsymbol{R}_t \in \mathbf{R}^{M\times M}$ be the correlation matrix and $\bar{\boldsymbol{Q}} \in \mathbf{R}^{M\times M}$ be a vector of unconditional variance.

Then, let $\boldsymbol{\varepsilon}_t \in \mathbf{R}^M$ be a vector of innovation term following iid.$N(\mathbf{0}, \boldsymbol{R}_t)$.

2.2 DCC-GARCH

$\boldsymbol{dr}_t$ can be represented by DCC(1,1)-GARCH(1,1) as follows:

$$\begin{aligned}
\boldsymbol{dr}_t &= \boldsymbol{\mu} + \boldsymbol{\sigma}_t \odot \boldsymbol{\varepsilon}_t, \\
\boldsymbol{\sigma}_t^2 &= \boldsymbol{\omega} + \boldsymbol{\alpha} \odot \boldsymbol{\sigma}_{t-1}^2 \odot \boldsymbol{\varepsilon}_{t-1} \odot \boldsymbol{\varepsilon}_{t-1} + \boldsymbol{\sigma}_{t-1}^2 \odot \boldsymbol{\beta}, \\
\boldsymbol{Q}_t &= (1-a-b)\bar{\boldsymbol{Q}} + a\boldsymbol{\varepsilon}_{t-1}\boldsymbol{\varepsilon}_{t-1}^{\mathrm{T}} + b\boldsymbol{Q}_{t-1}, \\
\boldsymbol{R}_t &= \mathrm{diag}\{\sqrt{\boldsymbol{Q}_t}\}^{-1}\boldsymbol{Q}_t\mathrm{diag}\{\sqrt{\boldsymbol{Q}_t}\}^{-1}.
\end{aligned}$$

where $\odot$ denotes the Hadamard product. The calculation method of the DCC-GARCH model parameters is the same as that in [5]. The all lag orders are assumed to be one, following [14] and [8].

2.3 Portfolio Loss

Let $D(t, T_m^{(i)})$ be the discount factor of $T_m^{(i)}$ year, defined by $D(t, T_m^{(i)}) = e^{-r_{T_m,t}(T_m^{(i)}-t)}$. Then, $f_t^{(i)}$ can be represented as follows:

$$f_t^{(i)} = \frac{D(t, T_s^{(i)}) - D(t, T_e^{(i)})}{\frac{1}{\psi}\sum_{d=\psi T_s^{(i)}+1}^{\psi T_e^{(i)}} D(t, \frac{d}{\psi})} = \frac{D(t, T_s^{(i)}) - D(t, T_e^{(i)})}{\text{Annuity}}.$$

where ψ is decided by the standard tenor. For example, in Japan, ψ is basically 2 (= 12 ÷ 6), since the standard tenor of JPY Libor is six months.

$\sigma_{f_t,N,t}$ can be represented as follows (see Appendix A for further details):

$$\sigma_{f_t} = \frac{\sqrt{\sum_{T_{k1},T_{k2}} A_{T_{k1}} A_{T_{k2}} R_{T_{k1},T_{k2},t}}}{\text{Annuity}}, \; T_{k1}, T_{k2} \in \{T_s, T_s + \frac{1}{\psi}, \cdots, T_e - \frac{1}{\psi}, T_e\}$$

where

$$A_{T_s} = -(T_s - t)D(t, T_j)\sigma_{T_s}, A_{T_e} = (1 + \frac{f_t}{\psi})(T_e - t)D(t, T_e)\sigma_{T_e},$$
$$A_{T_d} = \frac{f_t}{\psi}\sum_{T_d}(T_d - t)D(t, T_d)\sigma_{T_d}, T_d \in \{T_s + \frac{1}{\psi}, T_s + \frac{2}{\psi} \cdots T_e - \frac{2}{\psi}, T_e - \frac{1}{\psi}\}.$$

Most of the underlying of interest rate derivative products are forward swap rates, whereas the observed rate in the market is a par rate, such as a swap rate. Therefore, losses by various greeks need to be bucketed for all spot grids.

First, let $\boldsymbol{C_t}^{\text{del}}, \boldsymbol{C_t}^{\text{ve}} \in \mathbf{R}^M$ be vectors for representing the delta and vega risk in the respective grids; both of them can be represented as follows:

$$\boldsymbol{\phi}^{\text{del}(\boldsymbol{i})^{\text{T}}} = [\frac{df_t^{(i)}}{dr_{T_1,t}}, \frac{df_t^{(i)}}{dr_{T_2,t}}, \cdots, \frac{df_t^{(i)}}{dr_{T_M,t}}], \qquad \boldsymbol{C_t}^{\text{del}^{\text{T}}} = \sum_{i=1}^{I} \frac{dPV^{(i)}}{df_t^{(i)}} \boldsymbol{\phi}^{\text{del}(\boldsymbol{i})^{\text{T}}},$$
$$\boldsymbol{\phi}^{\text{ve}(\boldsymbol{i})^{\text{T}}} = [\frac{d\sigma_{f_t^{(i)}}}{d\sigma_{T_1,t}^S}\frac{1}{2\sigma_{T_1,t}^S}, \frac{d\sigma_{f_t^{(i)}}}{d\sigma_{T_2,t}^S}\frac{1}{2\sigma_{T_2,t}^S}, \cdots, \frac{d\sigma_{f_t^{(i)}}}{d\sigma_{T_M,t}^S}\frac{1}{2\sigma_{T_M,t}^S}], \quad \boldsymbol{C_t}^{\text{ve}^{\text{T}}} = \sum_{i=1}^{I} \frac{dPV^{(i)}}{d\sigma_{f_t^{(i)}}} \boldsymbol{\phi}^{\text{ve}(\boldsymbol{i})^{\text{T}}}.$$

Additionally, $\frac{df_t^{(i)}}{dr_{T_m,t}}, \frac{d\sigma_{f_t^{(i)}}}{d\sigma_{T_m,t}^S}$ are calculated by numerical differentiation.

Second, let $\boldsymbol{C_t}^{\text{gam}} \in \mathbf{R}^{M \times M}$ be a vector for representing the gamma risk in the respective grids. It can be represented as follows:

$$\boldsymbol{\phi}^{\text{gam}(\boldsymbol{i})} = \begin{bmatrix} \frac{df_t^{(i)}}{dr_{T_1,t}}\frac{df_t^{(i)}}{dr_{T_1,t}} & \cdots & \frac{df_t^{(i)}}{dr_{T_1,t}}\frac{df_t^{(i)}}{dr_{T_M,t}} \\ \vdots & \ddots & \vdots \\ \frac{df_t^{(i)}}{dr_{T_{M_1},t}}\frac{df_t^{(i)}}{dr_{T_1,t}} & \cdots & \frac{df_t^{(i)}}{dr_{T_M,t}}\frac{df_t^{(i)}}{dr_{T_M,t}}. \end{bmatrix}, \boldsymbol{C_t}^{\text{gam}} = \sum_{i=1}^{I} \frac{1}{2}\frac{d^2PV^{(i)}}{df_t^{(i)}df_t^{(i)}}\boldsymbol{\phi}^{\text{gam}(\boldsymbol{i})}.$$

The σ_{f_t} for valuing an option's present value is annualized generally, whereas $\boldsymbol{\sigma}_t$ on DCC-GARCH is on a daily basis. Therefore, $\boldsymbol{\sigma}_t$ needs to be annualized. More specific, the difference in the squared volatility per day is multiplied 250 business days, as follows:

$$\boldsymbol{d\sigma}^2 \approx (\boldsymbol{\sigma}_t^2 - \boldsymbol{\sigma}_{t-1}^2) \times 250.$$

Then, let dPV_t be defined as the portfolio loss using the above coefficients, $\boldsymbol{C_t^{\mathrm{del}}}, \boldsymbol{C_t^{\mathrm{gam}}}, \boldsymbol{C_t^{\mathrm{ve}}}$, as follows:

$$\begin{aligned}
\mathrm{dPV}_t &= \boldsymbol{C_{t-1}^{\mathrm{del}}}^{\mathrm{T}} \boldsymbol{dr}_t + \boldsymbol{dr}_t^{\mathrm{T}} \boldsymbol{C_{t-1}^{\mathrm{gam}}} \boldsymbol{dr}_t + \boldsymbol{C_{t-1}^{\mathrm{ve}}}^{\mathrm{T}} (\boldsymbol{\sigma}_t^2 - \boldsymbol{\sigma}_{t-1}^2) \times 250, \\
&= \boldsymbol{C}^{\mathrm{del}}{}_{t-1}^{\mathrm{T}} \boldsymbol{\mu}_t + \boldsymbol{\mu}_t^{\mathrm{T}} \boldsymbol{C}^{\mathrm{gam}}{}_{t-1} \boldsymbol{\mu}_t + \boldsymbol{C}^{\mathrm{ve}}{}_{t-1}^{\mathrm{T}} (\boldsymbol{\omega} + \mathrm{diag}\{\boldsymbol{\sigma}_t^2\}(\boldsymbol{\beta} - 1)) \times 250 \\
&+ (\boldsymbol{C}^{\mathrm{del}}{}_{t-1}^{\mathrm{T}} \mathrm{diag}\{\boldsymbol{\sigma}_{t-1}\} + 2\boldsymbol{\mu}_t^{\mathrm{T}} \boldsymbol{C}^{\mathrm{gam}}{}_{t-1} \mathrm{diag}\{\boldsymbol{\sigma}_{t-1}\}) \boldsymbol{\varepsilon}_t \\
&+ \boldsymbol{\varepsilon}_t^{\mathrm{T}} (\mathrm{diag}\{\boldsymbol{\sigma}_{t-1}\} \boldsymbol{C}^{\mathrm{gam}}{}_{t-1} \mathrm{diag}\{\boldsymbol{\sigma}_{t-1}\} + \boldsymbol{\alpha}^{\mathrm{T}} \mathrm{diag}\{\boldsymbol{\sigma}_{t-1}^2\} \mathrm{diag}\{\boldsymbol{C}^{\mathrm{ve}}{}_{t-1}\} \times 250) \boldsymbol{\varepsilon}_t.
\end{aligned}$$

Let $\boldsymbol{P_{t-1}} \in \mathbf{R}^M$ be the coefficient of the fourth term of $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{O}_{t-1} \in \mathbf{R}^{M \times M}$ be the coefficient of the fifth term $\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{\varepsilon}_t$. Then, the portfolio loss can be reformulated as follows:

$$\mathrm{dPV}_t = (\cdots) + \boldsymbol{P}_{t-1}^{\mathrm{T}} \boldsymbol{\varepsilon}_t + \boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{O}_{t-1} \boldsymbol{\varepsilon}_t. \tag{1}$$

2.4 Uncertainty Set

Let the following ellipse type be set as the uncertainty set:

$$\{\boldsymbol{\varepsilon}_t | \sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{R}_{t-1}^{-1} \boldsymbol{\varepsilon}_t} \leqq \delta \},$$

where $\sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{R}_{t-1}^{-1} \boldsymbol{\varepsilon}_t}$ is well known as the so-called "Mahalanobis distance"[1] and follows a chi-squared distribution with M degrees of freedom. The larger its value becomes, the larger the uncertainty of the overall risk factors becomes.

The set consists of all innovation terms and the correlation matrix. And the level of its value depends on the degree of scatter between each innovation term. More specific, the larger the correlation coefficient between interest rate of one grid and that of another grid is, the larger innovation terms of both is, as described in [15].

Here, $\sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{R}_{t-1}^{-1} \boldsymbol{\varepsilon}_t}$ is a nonlinear constraint equation and can be rewritten as a second-order cone. First, $\boldsymbol{R}_{t-1}^{-1}$ can be decomposed into $\boldsymbol{L}_{t-1} \boldsymbol{L}_{t-1}^{\mathrm{T}}$ by the Cholesky factorization. Referring to [4], let $\boldsymbol{z}_t$ be defined as follows:

$$\boldsymbol{z}_t = \boldsymbol{L}_{t-1}^{\mathrm{T}} \boldsymbol{\varepsilon}_t. \tag{2}$$

Next, $\sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{R}_{t-1}^{-1} \boldsymbol{\varepsilon}_t}$ can be reformulated as a second-order cone by (2), as follows:

$$\begin{aligned}
\sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{R}_{t-1}^{-1} \boldsymbol{\varepsilon}_t} &= \sqrt{\boldsymbol{\varepsilon}_t^{\mathrm{T}} \boldsymbol{L}_{t-1} \boldsymbol{L}_{t-1}^{\mathrm{T}} \boldsymbol{\varepsilon}_t} \\
&= \sqrt{\boldsymbol{z}_t^{\mathrm{T}} \boldsymbol{z}_t} \\
&= \|\boldsymbol{z}_t\|.
\end{aligned} \tag{3}$$

[1] The mean of the innovation terms is 0 in this formula.

Similarly, changeable parts in the portfolio loss (1) can also be reformulated using (2), as follows:

$$\boldsymbol{P}_{t-1}^{\mathrm{T}}(\boldsymbol{L}_{t-1}^{\mathrm{T}})^{-1}\boldsymbol{z}_t + \boldsymbol{z}_t^{\mathrm{T}}\boldsymbol{L}_{t-1}^{-1}\boldsymbol{O}_{t-1}(\boldsymbol{L}_{t-1}^{\mathrm{T}})^{-1}\boldsymbol{z}_t. \tag{4}$$

Here, let $\hat{\boldsymbol{P}}_{t-1} \in \mathbf{R}^M$ and $\hat{\boldsymbol{O}}_{t-1} \in \mathbf{R}^{M\times M}$ be defined as follows:

$$\hat{\boldsymbol{P}}_{t-1} = \boldsymbol{P}_{t-1}^{\mathrm{T}}(\boldsymbol{L}_{t-1}^{\mathrm{T}})^{-1}, \hat{\boldsymbol{O}}_{t-1} = \boldsymbol{L}_{t-1}^{-1}\boldsymbol{O}_{t-1}(\boldsymbol{L}_{t-1}^{\mathrm{T}})^{-1}. \tag{5}$$

Then, (4) can be reformulated using (5). Let $l(\boldsymbol{z}_t)$ be defined as the following function:

$$l(\boldsymbol{z}_t) = \hat{\boldsymbol{P}}_{t-1}^{\mathrm{T}}\boldsymbol{z}_t + \boldsymbol{z}_t^{\mathrm{T}}\hat{\boldsymbol{O}}_{t-1}\boldsymbol{z}_t. \tag{6}$$

2.5 Forecasting Scenario

The mathematical optimization programming can be constructed in order that forecast conceivable scenarios and significant losses, as follows:

$$\left| \begin{array}{ll} \min & l(\boldsymbol{z}_t) \\ \text{s.t.} & \|\boldsymbol{z}_t\| \leq \delta \quad \cdots (*) \end{array} \right. \tag{7}$$

(7) is nonconvex and nonlinear second-order cone programming because $\hat{\boldsymbol{O}}_t$ is not guaranteed to be a positive symmetric matrix. There exists no algorithm that is able to do so, in contrast to general convex second-order cone programming. Therefore, (7) is reformulated by introducing slack variables, as shown in the following theorem.

Theorem 1. *Let $\bar{\boldsymbol{z}}_t = (\delta; \boldsymbol{z}_t) \in \mathbf{R}^1 \times \mathbf{R}^M$ and $\bar{\boldsymbol{x}}_t = (x_{0,t}; \boldsymbol{x}_t) \in \mathbf{R}^1 \times \mathbf{R}^M$ be slack variables. Then, $(*)$ is equivalent to the following:*

$$\delta - \|\bar{\boldsymbol{x}}_t\|^2 = 0, \boldsymbol{z}_t - 2x_{0,t}\boldsymbol{x}_t = 0.$$

Proof. See [1]. □

Then, (7) can be reformulated nonlinear programming as follows:

$$\left| \begin{array}{ll} \min & l(\boldsymbol{z}_t) \\ \text{s.t.} & \delta - \|\bar{\boldsymbol{x}}_t\|^2 = 0, \boldsymbol{z}_t - 2x_{0,t}\boldsymbol{x}_t = 0. \end{array} \right. \tag{8}$$

By this approach, (8) is expected to be solved more efficiently than solving (7) directly because there exist some algorithms or solvers to solve nonlinear programming models. See [6] for a detailed theoretical explanation of this technique.

However, [15] pointed out that there exists some possibility that (8) cannot find appropriate solutions, because the sign of the solution would be inverse unintentionally. While [17] reported that outer approximation algorithm is a effective way to find appropriate solutions to concave quadratic function(8). Therefore, I apply the method of [17] to (8).

3 Outer Approximation

Roughly speaking, step of outer approximation algorithm are that reformulates to bilinear programming, finds local solutions and cuts feasible set repeatedly, and finally looks for the best solution. In this paper, it is normal that there is no difference on using this algorithm when compared to [17]. However, the constraint condition of target programming exists two differences when compared to that of [17].

One is to introduce the lower and upper bounds of the innovation term without existence in [17]. When there are their bounds, it needs a huge amount of calculation time and cut numbers to solve this programming. I guess because feasible set is so large. So, it needs to avoid such a case. Second, as in section 2.5, it includes not only linear inequality but also nonlinear equality in structure . This programming becomes complicated although, there is no a special issue in using this algorithm in the theory.

Let me explain detailed of outer approximation algorithm. $\hat{\boldsymbol{O}}_{t-1}$ can be decomposed by singular value factorization, since it is a real symmetric matrix, as follows:

$$\hat{\boldsymbol{O}}_{t-1} = \boldsymbol{d}\boldsymbol{\Lambda}\boldsymbol{d}^{\mathrm{T}}, \boldsymbol{d} = [\boldsymbol{d_1}, \cdots, \boldsymbol{d_M}].$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix whose elements are eigenvalues Λ_m of $\hat{\boldsymbol{O}}_{t-1}$ and $|\Lambda_1| \geqq \cdots \geqq |\Lambda_M|$. $\boldsymbol{d}$ is an orthonormal matrix, and $\boldsymbol{d_m} \in \mathbf{R}^M$ is a vector. Then, (6) can be reformulated as follows:

$$l(\boldsymbol{z}_t) = \hat{\boldsymbol{P}}_{t-1}^{\mathrm{T}} \boldsymbol{z}_t + \boldsymbol{z}_t^{\mathrm{T}} \boldsymbol{d}\boldsymbol{c}\boldsymbol{z}_t, \tag{9}$$

In addition, (9) can be bilinear programming, as follows:

$$\hat{\boldsymbol{P}}_{t-1}^{\mathrm{T}} \boldsymbol{z}_t + \hat{\boldsymbol{P}}_{t-1}^{\mathrm{T}} \boldsymbol{y}_t + 2\sum_{j=1}^{M} \boldsymbol{d_j}^{\mathrm{T}} \boldsymbol{y}_t \cdot \boldsymbol{c_j}^{\mathrm{T}} \boldsymbol{z}_t \tag{10}$$

Let $\underline{\boldsymbol{\varepsilon}}, \overline{\boldsymbol{\varepsilon}} \in \mathbf{R}^M$ be the lower and upper bounds of the innovation term $\boldsymbol{\varepsilon}$, and let the upper and lower bounds of $\boldsymbol{z}_t$ and $\boldsymbol{y}_t$ be defined as $\underline{\boldsymbol{z}}_t = \boldsymbol{L}_{t-1}^{\mathrm{T}} \underline{\boldsymbol{\varepsilon}}, \overline{\boldsymbol{z}} = \boldsymbol{L}_{t-1}^{\mathrm{T}} \overline{\boldsymbol{\varepsilon}}$ and $\underline{\boldsymbol{y}}_t = \boldsymbol{L}_{t-1}^{\mathrm{T}} \underline{\boldsymbol{\varepsilon}}, \overline{\boldsymbol{y}} = \boldsymbol{L}_{t-1}^{\mathrm{T}} \overline{\boldsymbol{\varepsilon}}$.

Here, let $\mathbf{Z}$ and $\mathbf{Y}$ be a feasible set defined as follows:

$$\begin{aligned} \mathbf{Z} &= \{\boldsymbol{z}_t \in \mathbf{R}^M | \delta - \|\bar{\boldsymbol{x}}_t{}^{(z)}\|^2 = 0, \boldsymbol{z}_t - 2x_{0,t}^{(z)} \boldsymbol{x}_t^{(z)} = 0, \underline{\boldsymbol{z}}_t \leqq \boldsymbol{z}_t \leqq \overline{\boldsymbol{z}}_t\}, \\ \mathbf{Y} &= \{\boldsymbol{y}_t \in \mathbf{R}^M | \delta - \|\bar{\boldsymbol{x}}_t{}^{(y)}\|^2 = 0, \boldsymbol{y}_t - 2x_{0,t}^{(y)} \boldsymbol{x}_t^{(y)} = 0, \underline{\boldsymbol{y}}_t \leqq \boldsymbol{y}_t \leqq \overline{\boldsymbol{y}}_t\}. \end{aligned} \tag{11}$$

These feasible sets are the main difference when compare to [17].

Let me introduce the $M+1$ auxiliary variable $\boldsymbol{\xi} = [\xi_0, \cdots, \xi_M]^{\mathrm{T}} = [\hat{\boldsymbol{P}}_{t-1}, \boldsymbol{d}]^{\mathrm{T}} \boldsymbol{y}_t = \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{y}_t \in \mathbf{R}^{M+1}$ and define the following problem:

$$\left| \begin{array}{ll} \min & \boldsymbol{c}_0^{\mathrm{T}} \boldsymbol{z}_t + \xi_0 + 2\sum_{j=1}^{M} \xi_j \cdot \boldsymbol{c}_j^{\mathrm{T}} \boldsymbol{z}_t \\ \text{s.t.} & \boldsymbol{z}_t \in \mathbf{Z}, \boldsymbol{y}_t \in \mathbf{Y}, \boldsymbol{\xi} = \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{y}. \end{array} \right. \tag{12}$$

(12) is relaxed, and it is solved repeatedly using an outer approximation algorithm by following [17]. First, let $\underline{\xi_j}$ and $\overline{\xi_j}$ be defined as:

$$\begin{aligned} \underline{\xi_j} &= \min\{\xi_j | (\xi_0, \cdots, \xi_M) \in \mathbf{G}^k\}, j = 0, 1, \cdots, M; k = 0, 1, \cdots, \\ \overline{\xi_j} &= \max\{\xi_j | (\xi_0, \cdots, \xi_M) \in \mathbf{G}^k\}, j = 0, 1, \cdots, M; k = 0, 1, \cdots. \end{aligned}$$

Second, let $\mathbf{G}^0$ be defined as:

$$\mathbf{G}^0 = \{\boldsymbol{\xi} \in \mathbf{R}^{M+1} | \underline{\xi_i} \leqq \xi_j \leqq \overline{\xi_i}, j = 0, 1, \cdots, M\}$$

Let me define the $k(0, 1, \cdots)$th relaxation problem. $\xi^k_{j,BP}, \boldsymbol{z}^k_{BP}$ can be obtained as the solution of the following bilinear problem:

$$(BP)^k \left| \begin{array}{ll} \min & \boldsymbol{c}_0^{\mathrm{T}} \boldsymbol{z}_t + \xi_0 + 2\sum_{j=1}^{M} \xi_j \cdot \boldsymbol{c}_j^{\mathrm{T}} \boldsymbol{z}_t \\ \text{s.t.} & \boldsymbol{z}_t \in \mathbf{Z}, \boldsymbol{\xi} \in \mathbf{G}^k. \end{array} \right. \tag{13}$$

To make $\boldsymbol{\xi} = \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{y}$ realized, $\xi^k_{j,LSM}, \boldsymbol{z}^k_{LSM}$ can be obtained as the solution of the following convex quadratic problem:

$$(LSM)^k \left| \begin{array}{ll} \min & \sum_{j=0}^{M} (\xi_j - \xi^k_{j,BP})^2 \\ \text{s.t.} & \boldsymbol{y}_t \in \mathbf{Y}, \boldsymbol{\xi} = \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{y}. \end{array} \right. \tag{14}$$

Let me define the following affine function:

$$H^k(\boldsymbol{\xi}) = \sum_{j=0}^{M} (\xi^k_{j,LSM} - \xi^k_{j,BP})(\xi_j - \xi^k_{j,LSM}) \tag{15}$$

$\mathbf{G}^{k+1}$ can then be constructed by adding a cut:

$$\mathbf{L}^k = \{\boldsymbol{\xi} \in \mathbf{R}^{M+1} | H^k(\boldsymbol{\xi}) \geqq 0\}, \mathbf{G}^{k+1} = \mathbf{G}^k \cap \mathbf{L}^k.$$

(15) is a hyperplane function. Therefore, by repeating this, $\mathbf{G}^{k+1}$ can create convex polyhedral sets, because a feasible set can be cut off by cutting plane repeatedly. In summary, the algorithm is as follows.

Outer Approximation Algorithm

Step1 Let k=0.

Step2 Compute a solution ξ^k_{BP} of the relaxation problem $(BP)^k$.

Step3 Construct the cut $\mathbf{L}^k$ by (14) and (15).

Step4 If $||\boldsymbol{\xi}^k_{BP} - \boldsymbol{\xi}^k_{LSM}|| \leqq \eta$, then stop. Otherwise, let

$$\mathbf{G}^{k+1} = \mathbf{G}^k \cap \mathbf{L}^k, k := k + 1.$$

and return to Step2.

Theorem 2. *The algorithm generates a sequence of points $\boldsymbol{\xi}^k$ which converge to a global solution $\boldsymbol{\xi}^*$.*

Proof. See [10]. □

Lastly, $\hat{\boldsymbol{\varepsilon}}_{LSM}, \hat{\boldsymbol{z}}_{LSM}, \hat{\boldsymbol{\varepsilon}}_{BP}$ and $\hat{\boldsymbol{z}}_{BP}$ can be obtained. Additionally, see Appendix B for more details on how to solve at Step2.

In [17], coefficient of bilinear programming were made by uniform random variable. While, I make using portfolio data with high of nonlinear risk. Here, nonlinear risk means vega mainly. As vega become short(negative), degree of concavity of object function be higher. And then, each diagonal components of $\hat{\boldsymbol{O}}$ become negative and object function become concave.

Also, main business of investment bank and security company is to sell various financial products, so vega of their portfolios tend to be short/negative inevitably. Therefore, I focus the strength case of concavity from the perspective of financial risk management.

4 Results and Analysis

4.1 Setting

The JPY swap rate is used as the market rate; therefore, let ψ be 2. A linear interpolation is used to interpolate the JPY swap rate when calculating the discount factor (D). Let $T_m^{\mathrm{T}} = \{1,2,3,4,5,6,7,8,9,10,12,15,20,30\}$. Then, $\frac{dPV^{(i)}}{df_t^{(i)}}$, $\frac{d^2PV^{(i)}}{df_t^{(i)}df_t^{(i)}}$, and $\frac{dPV^{(i)}}{d\sigma_{f_t}^{(i)}}$ are calculated using the normal model for valuing european swaption (see Appendix C for further details).

Also, $\frac{df_t^{(i)}}{dr_{T_m,t}}$ is calculated by numerical differentiation.

I prepared forty four portfolios for calculation experiment, which are made by choosing randomly on the following restrictions:

- Notional: The notional limit is 10 JPY (yen) per trade.
- $T_k^{(i)}$:$k \in \{2,3,...,13,14\}$ as maturity of trade i.
- $T_j^{(i)}$: $j \in \{1,...,T_k^{(i)}\}$ as option term of trade i..
- Strike$^{(i)}$: $f_t^{(i)} + (n-7) \times \frac{|f_t^{(i)}|}{4}$, where "n" is chosen randomly in {1,2,..., 12,13}.
- I:3000

The trade count for notional, $T_k^{(i)}$ and $T_k^{(i)}$, and strike(n) in the example of one portfolio are as follows.

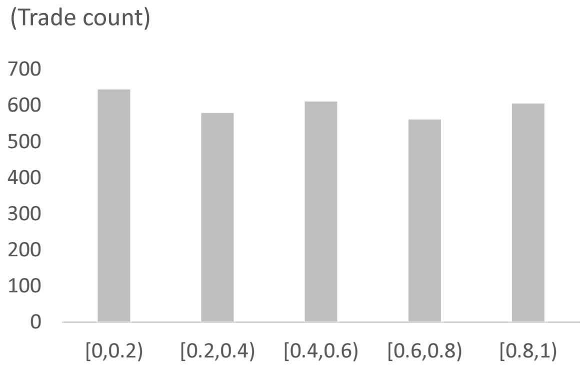

Figure 1: Histogram of the trade count for each notional

And, vega per 1bp [2] and $\boldsymbol{O}$ in same example are as follows.

Table 2: Example of vega per 1bp

grid	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	12Y	15Y	20Y	30Y
vega	0	0	2	4	8	-1	-2	-5	15	-6	-54	-8	35	-13

[2] I omit gamma since this greek is not so important.

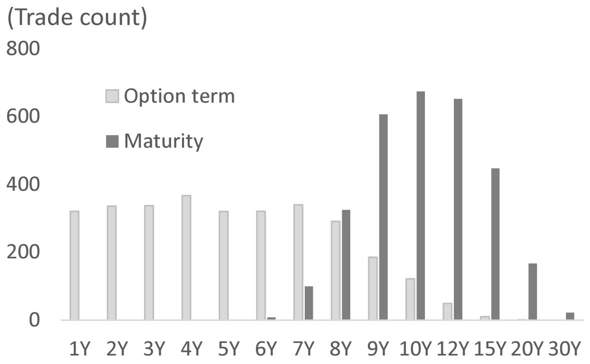

Figure 2: Histogram of trade count for each option term and maturity

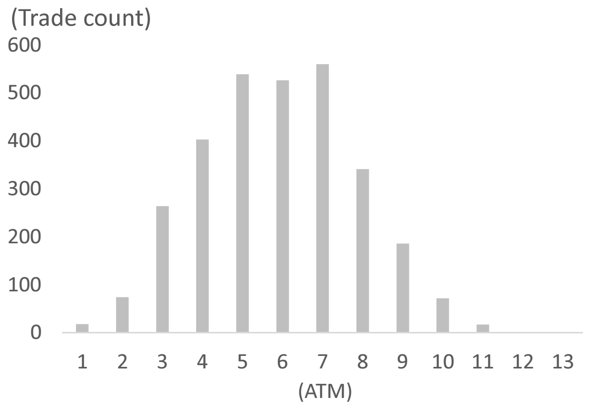

Figure 3: Histogram of trade count for each strike

Table 3: Example of $\boldsymbol{O}$

grid	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	12Y	15Y	20Y	30Y
1Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2Y	0	1	0	0	0	0	0	0	0	0	0	0	0	0
3Y	0	0	8	0	0	0	0	0	0	0	0	0	0	0
4Y	0	0	0	10	0	0	0	0	0	0	0	0	0	0
5Y	0	0	0	0	23	0	0	0	0	0	0	0	0	0
6Y	0	0	0	0	0	-4	2	1	-1	0	0	-1	0	0
7Y	0	0	0	0	0	2	-6	-3	1	0	0	0	0	0
8Y	0	0	0	0	0	1	-3	-13	-3	-1	2	0	0	0
9Y	0	0	0	0	0	-1	1	-3	73	-2	1	0	0	0
10Y	0	0	0	0	0	0	0	-1	-2	-39	1	1	0	0
12Y	0	0	0	0	0	0	0	2	1	1	-426	1	0	0
15Y	0	0	0	0	0	-1	0	0	0	1	1	-77	0	0
20Y	0	0	0	0	0	0	0	0	0	0	0	0	432	0
30Y	0	0	0	0	0	0	0	0	0	0	0	0	0	-184

Negative of vega in 10Y, 12Y, 15Y and 30Y in table 2, and negative of diagonal components in same grid in table 3. Also, it can guess that this matrix is no positive definite matrix because there is negative more than one count in diagonal components.

In fact, it exists negative in eigen values of $\hat{\boldsymbol{O}}$ as follows.

$$0.024 \geq 0.013 \geq 0.001 \geq 0 \geq 0 \geq 0 \geq 0 \geq 0 \geq 0 \geq 0 \geq 0 \geq 0 \geq -0.006 \geq -0.23$$

As above mentioned, a point is in the case of strength of concavity of object function. However, I don't know a best gauge for measuring degree of concavity of object function of rank fourteen all[3]. So, I make the following original gauge $\hat{\lambda}$.

$$\hat{\lambda} = \frac{|\min \lambda_m|}{|\max \lambda_m|}. \tag{16}$$

I intend that the most convex dimension of fourteen dimensions is compared with the most concave dimension of them. As this value become larger, concavity of object function become stronger. So, I aim that $\ln(\hat{\lambda})$ shows degree of concavity of object function of rank fourteen all to some extent. $\ln(\hat{\lambda})$ of forty four portfolio are as follows.

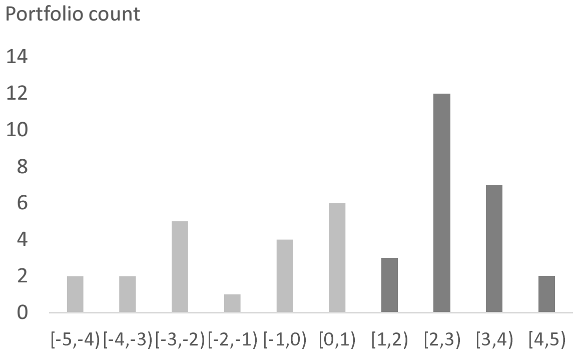

Figure 4: Histogram of $\ln(\hat{\lambda})$ for each portfolio

In terms of twenty four out of forty four, $\ln(\hat{\lambda})$ become larger than one, they are of especial interesting material for me, since degree of concavity of them is so large.

Let the observation period be 501 business days when calculating the DCC-GARCH parameters and the future period be 1 business day. The initial value of $\boldsymbol{z}_t$ is $\mathbf{0} \in \mathbf{R}^M$. δ, $\overline{\varepsilon}$, $\underline{\varepsilon}$ and η are used for the following setting. Furthermore, $\underline{\varepsilon}$ is equivalent to $-\overline{\varepsilon}$.

Table 4: Parameter settings

δ	1.089, 1.3307, 1.5893, 1.874
$\overline{\varepsilon}(\underline{\varepsilon})$	2,3,4,5
η	$1.0 \times 10^{-2}, 5.0 \times 10^{-3}, 1.0 \times 10^{-3}$

The program was coded in R and run on a personal computer (Core i7-7500, 8.0 GB, 2.70 GHz). However, only parts of the numerical differences were coded in C++ by the "Rcpp" module of R. Furthermore, the "rmgarch" module [7] of R was used to calculate the DCC-GARCH parameters, the "alabama" module [13] of R was used to solve (13), and the "cccp" module [2] of R was used to solve (14).

[3] I guess there is no in the world.

4.2 Computational Result

In figure 5, the horizontal axis represents $\ln(\hat{\lambda})$, and the vertical axis represents the relative error between $f(\hat{\boldsymbol{\varepsilon}}_{LSM}, \hat{\boldsymbol{z}}_{LSM})$ and $f(\hat{\boldsymbol{\varepsilon}}_{BP}, \hat{\boldsymbol{z}}_{BP})$ [4]. When the relative error would be nearly zero, the convergence of outer approximation algorithm is better. Figure 5 shows that the accuracy of the algorithm is better as $\hat{\lambda}$ become larger. Therefore, this algorithm is expected to solve more efficiently as degree of concavity of the objective function become stronger. Conversely, this algorithm should not be used when degree of concavity of the objective function is so small [5]. Considering this result, as for overall forty four portfolios, calculation target after here are limited to twenty four portfolios which $\ln(\hat{\lambda})$ become larger than one or $\hat{\lambda}$ become larger than zero.

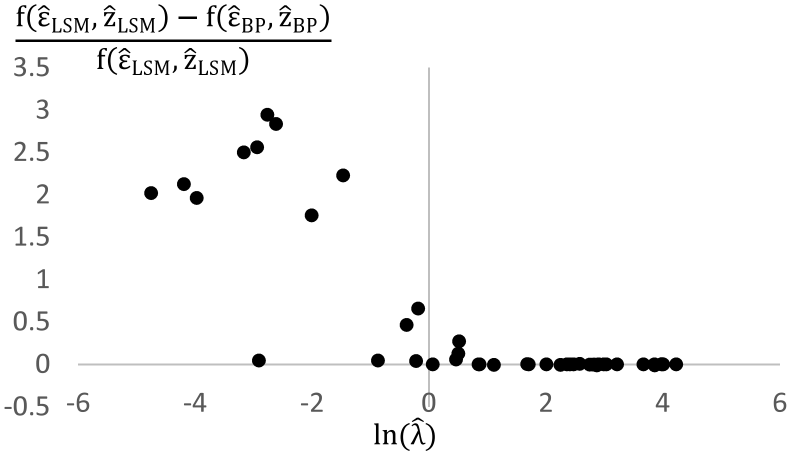

Figure 5: Relative error of solutions of (BP) and (LSM)

Figures 6 and 7 show the relationship between the strength of concavity and the calculation time/the cut number. I see from their figures that the calculation cost become smaller as degree($\hat{\lambda}$) of the concavity become larger. In fact, the correlation coefficient between the calculation time or the cut number and $\hat{\lambda}$ are -0.527 or -0.419.

From here, I'd like to think appropriate level of η and $\overline{\varepsilon}$. In general, there exists a trade-off relationship between increasing calculation cost and improving calculation accuracy. It is important to decide them, because degree of these parameter linked calculation cost and calculation accuracy directly,

Firstly, table 5 shows the average and the standard deviation (number in the bracket) of the calculation time and table 6 shows the average and the standard deviation (number in the bracket) of the cut number when $\eta, \overline{\varepsilon}$ and δ change. I see that, basically, the larger their parameters are, the larger the calculation time and the cut number are. Especially, impact by η is the largest. Calculation time in $\eta = 1.0 \times 10^{-2}$ is a few times shorter than that in $\eta = 5.0 \times 10^{-2}$ and is about tens times shorter than in $\eta = 1.0 \times 10^{-3}$. So, I think that calculation cost in the case of $\eta = 1.0 \times 10^{-3}$ is so heavy.

Thus, I check calculation accuracy when η change. Table 7 shows the average for

[4] It means that $\frac{f(\hat{\boldsymbol{\varepsilon}}_{LSM}, \hat{\boldsymbol{z}}_{LSM}) - f(\hat{\boldsymbol{\varepsilon}}_{BP}, \hat{\boldsymbol{z}}_{BP})}{f(\hat{\boldsymbol{\varepsilon}}_{LSM}, \hat{\boldsymbol{z}}_{LSM})}$.

[5] In this case, (8) is expected to be solved by using some general nonlinear solvers.

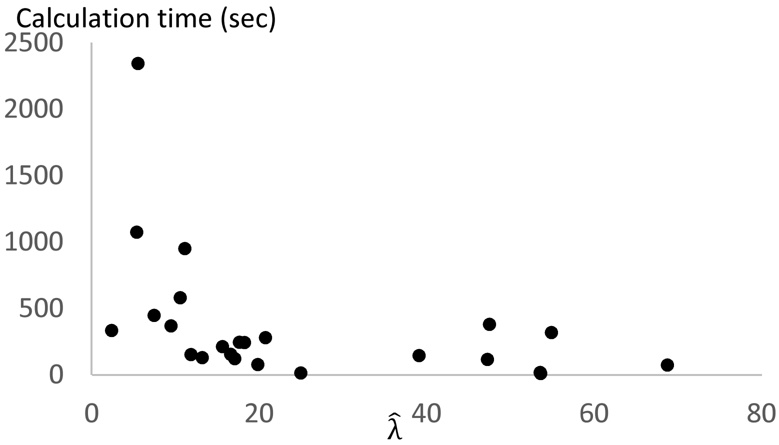

Figure 6: Calculation time for each $\hat{\lambda}$ in the case of $\overline{\varepsilon} = 4, \delta = 1.3307$ and $\eta = 5.0 \times 10^{-2}$

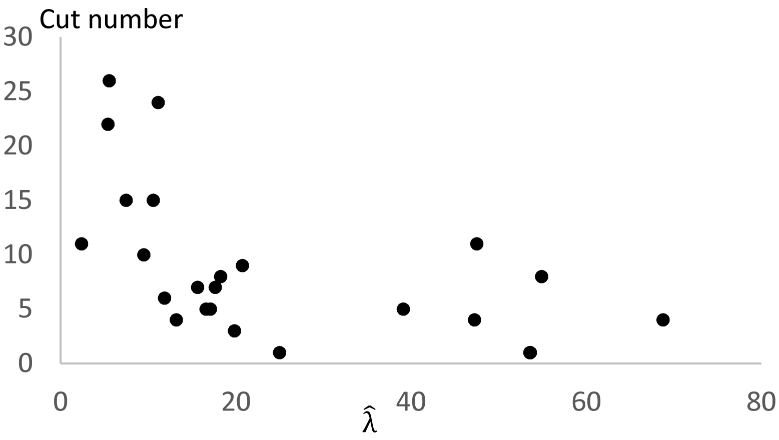

Figure 7: Cut number for each $\hat{\lambda}$ in the case of $\overline{\varepsilon} = 4, \delta = 1.3307$ and $\eta = 5.0 \times 10^{-2}$

Table 5: Calculation time (sec)

η	1.0×10^{-2}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	97.84(77.04)	100.75(81.78)	109.71(96.39)	118.54(103.66)
3	210.8(272.79)	238.73(423.42)	190.61(190.5)	221.65(228.88)
4	183.23(215.51)	282.37(333.47)	265.48(284.07)	268.9(306.59)
5	273.41(352.03)	237.86(294.43)	274.88(383.67)	270.5(303.16)
η	5.0×10^{-2}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	161.69(116.06)	167.91(138.78)	201.84(231.21)	197.65(240.81)
3	341.66(459.6)	365.67(519.08)	269.56(272.08)	277.04(354.8)
4	328.67(415.21)	367.24(499.19)	398.38(419.58)	497.6(632.14)
5	418.08(486.21)	516.19(675.4)	565.03(719.14)	792.56(2002.22)
η	1.0×10^{-3}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	900.19(1359.45)	888.74(1469.71)	594.31(805.62)	622.78(1014.38)
3	875.22(993.32)	1616.66(2511.57)	1999.25(3640.47)	1938.08(3399.05)
4	1032.42(1040.04)	1501.2(1759.24)	3177.51(5093.22)	6506.72(13691.51)
5	3638.07(5074.6)	3574.61(5368.72)	5845.44(7178.1)	3699.71(7983.92)

Table 6: Cut number

η	1.0×10^{-2}			
$\overline{\boldsymbol{\varepsilon}}$/ δ	1.089	1.3307	1.5893	1.874
2	3.38(2.07)	3.96(2.62)	3.84(2.59)	3.8(2.66)
3	6.38(6.75)	6.38(6.81)	5.71(4.21)	6.25(5.24)
4	5.8(5.24)	7.42(6.83)	7.55(6.88)	7.59(6.97)
5	7.75(7.62)	8.17(7.81)	8.67(9.05)	7.42(7.43)
η	5.0×10^{-2}			
$\overline{\boldsymbol{\varepsilon}}$/ δ	1.089	1.3307	1.5893	1.874
2	5.05(2.78)	5.17(2.98)	5.55(4.48)	4.96(4.17)
3	8.38(8.01)	8.92(8.05)	7.38(5.61)	8.3(7.74)
4	7.88(6.33)	8.84(7.02)	9.25(7.14)	10.21(8.78)
5	9.8(7.69)	10.42(9.17)	10.92(10.27)	10.46(13.2)
η	1.0×10^{-3}			
$\overline{\boldsymbol{\varepsilon}}$/ δ	1.089	1.3307	1.5893	1.874
2	14.55(12.78)	15.13(13.61)	12.42(9.35)	12(10.75)
3	14.63(10.25)	17.75(14.43)	20.25(16.71)	20.34(18.24)
4	18.84(10.41)	19(10.84)	25.13(22.97)	33.96(37.9)
5	29.88(18.2)	25.21(17.81)	34.88(24.19)	26.88(32.38)

all cases [6] of relative error[7] between the objective function values in $\eta = 1.0 \times 10^{-2}$ or 5.0×10^{-3} and the objective function values in $\eta = 1.0 \times 10^{-3}$. I see that calculation accuracy is better, as η decreases. However, considering this result as well as burden of calculation cost in table 5 and 6 when η changes, I think that $\eta = 5.0 \times 10^{-2}$ is appropriate.

Table 7: Average of relative value of objective function value on η (unit:10^{-4})

η	1.0×10^{-2}	5.0×10^{-3}
Relative error	27.83	12.52

Next, I check calculation accuracy when $\overline{\boldsymbol{\varepsilon}}$ changes. Table 8 shows the average of the relative value[8] between value in the case of $\overline{\boldsymbol{\varepsilon}} = 2, 3, 4$ and that in the case of $\overline{\boldsymbol{\varepsilon}} = 5$ when each parameter changes. As this relative value become close to zero, calculation accuracy become better. It is because that solution($\hat{\boldsymbol{\varepsilon}}_{LSM}, \hat{\boldsymbol{z}}_{LSM}$) in $\overline{\boldsymbol{\varepsilon}} = 5$ become the best so that could be obtained suitable solution since feasible set is the largest in the case of $\overline{\boldsymbol{\varepsilon}} = 2, 3, 4$ and 5.

I see that the calculation accuracy becomes worse as $\overline{\boldsymbol{\varepsilon}}$ becomes smaller and δ increase. It should avoid to choose $\overline{\boldsymbol{\varepsilon}} = 2$, because relative error is over 100bp. And I check the case where relative error isn't 50bp. Thus, I think that $\overline{\boldsymbol{\varepsilon}} = 3$ in the case of $\delta = 1.089$ or 1.3307 is and $\overline{\boldsymbol{\varepsilon}} = 4$ in the case of $\delta = 1.5893$ or 1.874 is best.

Lastly, I'd like to investigate appropriate level of δ. Table 9 shows the change range of hypothetical made by the model and actual change range in Japan market. The change range forecasted by the model in 1 day is multiplied 20 business days hypothetically. This change range of hypothetical in 20 business days for each δ is compared with change range

[6] 24 (portfolio) × 4 ($\overline{\boldsymbol{\varepsilon}}$)× 4 ($\delta$) =384.

[7] It means that $\frac{f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\eta=1.0\times10^{-2}\text{or } 5.0\times10^{-3})-f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\eta=1.0\times10^{-3})}{f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\eta=1.0\times10^{-3})}$.

[8] It means that $\frac{f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\overline{\boldsymbol{\varepsilon}}=2,3\text{or}4)-f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\overline{\boldsymbol{\varepsilon}}=5)}{f(\hat{\boldsymbol{\varepsilon}}_{LSM},\hat{\boldsymbol{z}}_{LSM}|\overline{\boldsymbol{\varepsilon}}=5)}$

Table 8: Relative error of objective function on $\overline{\varepsilon}$(unit:%)

η	1.0×10^{-2}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	-0.45	-0.87	-1.22	-1.55
3	-0.12	-0.36	-0.58	-0.81
4	0.16	-0.08	-0.09	-0.34
η	5.0×10^{-2}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	-0.60	-0.90	-1.15	-1.44
3	-0.21	-0.37	-0.46	-0.93
4	-0.02	-0.04	-0.07	-0.22
η	1.0×10^{-3}			
$\overline{\varepsilon}/\ \delta$	1.089	1.3307	1.5893	1.874
2	-0.64	-1.01	-1.14	-1.23
3	-0.23	-0.47	-0.47	-0.54
4	-0.03	-0.20	-0.15	-0.04

of the actual Japan market from January 2016 to 26th February 2016. Also, the reason I choose this periods is why the Bank of Japan announced introducing of a negative interest rate on 29th January 2016 and then interest rate market in Japan moved drastically.

As long as I see this result, in particular some grid case from 9Y to 15Y gathered vega risk as shown in section 4.1, it needs a magnitude from 1.089 to 1.3307 at least in order that represent a conceivable scenario in the case of high risk, similar to the announcement of a negative interest rate in Japan market.

Table 9: Change range of hypothetical for each δ and actual in the case of $\overline{\varepsilon} = 5$ (unit:bp)

δ/grid	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	12Y	15Y	20Y	30Y
1.089	2	4	3	8	10	13	17	20	23	25	26	30	41	49
1.3307	2	4	4	9	12	16	20	23	26	29	30	35	44	52
1.5893	2	4	5	11	14	19	24	26	30	33	34	40	49	58
1.874	2	3	6	13	16	22	27	30	34	37	38	45	55	65
Actual	13	22	24	25	25	25	25	25	26	27	28	30	33	34

Above all, it needs the level of δ from 1.089 to 1.3307 in order that represent high risk scenario similar to the scene of introducing of negative interest rate, and then requires $\overline{\varepsilon} = 3$ and $\eta = 5.0 \times 10^{-2}$ from the perspective of calculation accuracy and cost.

5 Conclusion and Future Research Directions

I showed that outer approximation algorithm is a effective calculation step in the case of very high of nonlinear risk, in particular short vega i.e. strong of concavity of object function. Actual financial institution tend to have high short vega and then it leads to be stronger of concavity of object function i.e. negative of diagonal components of $\boldsymbol{O}(\hat{\boldsymbol{O}})$. In this time, I find relationship of negative coefficient between strength of concavity of object function and calculation cost. So, the algorithm is very helpful from the perspective of efficient calculation and risk management.

I raise one future research. It is in the case of higher of nonlinear risk including vega as well as vanna. Vanna are second order differential i.e. cross gamma by shown movement both of interest rate and volatility. So, it is not easy to handle vanna in forecasting losses and scenarios, different from handling one order differential such as delta, gamma and vega by shown movement of either interest rate or volatility in computation.

While, in the case of huge of nonlinear risk of portfolio[9], it should investigate how to forecast losses and scenarios more accurately when capturing vanna in risk management.

Appendix A Variance of the Forward Swap Rate

First, referring to [3],we have:

$$\left|\begin{aligned} dr_{t,T_m} &= \mu_{T_m} + \sigma^S_{T_m,t}\varepsilon_{t,T_m}\sqrt{\tau}, \\ D(t,T_m) &= e^{A(t,T_m)-B(t,T_m)r_{t,T_m}}, \\ A(t,T_m) &= \frac{\sigma}{2}\frac{(T_m-t)^3}{3} + \mu_{T_d}\frac{(T_m-t)^2}{2}, \\ B(t,T_m) &= T_m - t. \end{aligned}\right.$$

where τ refers to a time step and $BD(t,T_m) = B(t,T_m)D(t,T_m)$.

Second, $df_t^{(i)}$ can be represented by using Ito's formula and referring to [16], as follows:

$$\begin{aligned} df_t &= \frac{\{dD(t,T_s) - \frac{f_t}{\psi}\sum_{T_d} dD(t,T_d) - (1+\frac{f_t}{\psi})dD(t,T_e)\}}{\text{Annuity}}, \\ &= \frac{\{-BD(t,T_s)dr_{t,T_s} + \frac{f_t}{\psi}\sum_{T_d} BD(t,T_d)dr_{t,T_d} + (1+\frac{f_t}{\psi})BD(t,T_e)dr_{t,T_e}\}}{\text{Annuity}}, \\ &= (\cdots) + \frac{\{-BD(t,T_s)\sigma_{T_s}\varepsilon_{T_s} + \frac{f_t}{\psi}\sum_{T_d} BD(t,T_d)\sigma_{T_d}\varepsilon_{T_d} + (1+\frac{f_t}{\psi})BD(t,T_e)\sigma_{T_e}\varepsilon_{T_e}\}}{\text{Annuity}}\sqrt{\tau}, \end{aligned}$$

The variance of the forward swap rate is given as follows:

$$\sigma^2_{f_t} = \frac{\sum_{T_{k1},T_{k2}} A_{T_{k1}}A_{T_{k2}}R_{T_{k1},T_{k2},t}}{\text{Annuity}^2},\ T_{k1},T_{k2} \in \{T_s, T_s+\frac{1}{\psi},\cdots,T_e-\frac{1}{\psi},T_e\}$$

where

$$A_{T_s} = -BD(t,T_s)\sigma_{T_s}, A_{T_d} = \frac{f_t}{\psi}\sum_{T_d} BD(t,T_d)\sigma_{T_d}, A_{T_e} = (1+\frac{f_t}{\psi})BD(t,T_e)\sigma_{T_e},$$

$$T_d \in \{T_s+\frac{1}{\psi}, T_s+\frac{2}{\psi}\cdots T_e-\frac{2}{\psi}, T_e-\frac{1}{\psi}\}.$$

Appendix B Details at Step2 of Outer Approximation Algorithms

Let me introduce the slack variable vector $\boldsymbol{\phi}$ and the all variable set $\Omega \in \{\bar{\boldsymbol{x}}_t{}^{(z)}, \boldsymbol{z}_t, \boldsymbol{\xi}, \boldsymbol{\phi}\}$. I distinguish the objective function $f(\Omega)$, the equality constraints $h(\Omega)$, and the inequality

[9] It is in the case of having a lot of exotic product such as trigger, digital, barrier, and bermudan-type.

constraints $g(\Omega)$ in (13) as follows:

$$\begin{aligned} h(\Omega) &= \begin{cases} \delta - \|\bar{\boldsymbol{x}_t}^{(z)}\|^2, \boldsymbol{z}_t - 2x_{0,t}^{(z)}\boldsymbol{x}_t^{(z)}, \\ \boldsymbol{z}_t - \underline{\boldsymbol{z}}_t - \phi_j, j = 1, \cdots, M, \\ \overline{\boldsymbol{z}}_t - \boldsymbol{z}_t - \phi_j, j = M+1, \cdots, 2M \\ \boldsymbol{\xi} - \underline{\boldsymbol{\xi}} - \phi_j, j = 2M+1, \cdots, 3M+1, \\ \overline{\boldsymbol{\xi}} - \boldsymbol{\xi} - \phi_j, j = 3M+2, \cdots, 4M+2, \\ H(\boldsymbol{\xi}) + \phi_j, j = 4M+3, \cdots 4M+3+k, \end{cases} \\ g(\Omega) &= \boldsymbol{\phi}. \end{aligned}$$

(13) can be an augmented Lagrangian function, according to [11] and [12], as follows:

$$\left| \begin{array}{ll} \min & f(\Omega) - \boldsymbol{\kappa}^{\mathrm{T}} h(\Omega) + \frac{\Sigma}{2}||h(\Omega)|| - \rho(g(\Omega^*)^{\mathrm{T}} \log(\Omega) - \frac{dg(\Omega^*)}{d\Omega^*}^{\mathrm{T}}\Omega) \\ \text{s.t.} & h(\Omega) = \mathbf{0}, g(\Omega) \geqq \mathbf{0}. \end{array} \right. \tag{17}$$

(17) can be solved by using the “constrOptim.nl” function of the “alabama” package in R. In this package, the Σ for equality constraints increases to ∞ and ρ as the barrier penalty for the inequality constraints decreases to 0 sequentially with refreshing of the scaling parameter Σ and ρ by solving repeatedly. In addition, Ω^* refers to the previous solutions, and $\boldsymbol{\kappa}$ represents the Lagrange multiplier vector.

In (17), the initial value of Σ is used (0.01) and that of ρ is used (10). However, when $\|\hat{\boldsymbol{z}}_{BP}\| \leq \delta - 10^{-5}$, the initial value of ρ is changed $\rho = \rho + 1$, and recalculation is conducted.

Appendix C Analytic Formula of Normal Model

The following table refers to [9].

PV	Annuity$\{\epsilon(f - \text{Strike})N(\epsilon d) + \sigma_f\sqrt{T_s}n(d)\}$
Delta	Annuity$\{\epsilon N(\epsilon d)\}$
Gamma	Annuity$\{\frac{n(d)}{\sigma_{f_t}\sqrt{T_s}}\}$
Vega	Annuity$\{\sqrt{T_s}n(d)\}$

Here, $d = \frac{f_t - \text{Strike}}{\sigma_{f_t}\sqrt{T_s^{(i)}}}$, $\epsilon = 1$ for payers and $\epsilon = -1$ for receivers. $n(\cdot)$ is the standard normal density function, and $N(\cdot)$ is the standard normal cumulative function.

References

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2003.

[2] P. Bernhard. Package “ cccp ”. `https://cran.r-project.org/web/packages/cccp/cccp.pdf`, 2015.

[3] T. Björk. *Arbitrage Theory in Continuous Time(3rd ed.).* Oxford university press, 2009.

[4] S. Boy and L. Vandenberghe. *Convex Optimization.* Cambridge University., 2004.

[5] R. F. Engle. Dynamic conditional correlation: a simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business and Economic Statistics*, 20(3):339–350, 2002.

[6] E. H. Fukuda and M. Fukushima. The use of squared slack variables in nonlinear second-order cone programming. *Journal of Optimization Theory and Applications*, 170(2):394–418, 2016.

[7] A. Ghalanos. Package “ rmgarch ”. `http://cran.r-project.org/web/packages/rmgarch/rmgarch.pdf`, 2015.

[8] P.R. Hansen and A. Lundle. A forecast comparison of volatility models:does anything beat a garch(1,1)? *Journal of applied econometrics*, 20:873–889, 2005.

[9] E. G. Haug. *The Complete Guide to Option Pricing Formulas (2nd edition).* McGraw-Hill Education, 2007.

[10] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches.* Springer-Verlag Berlin Heidelberg, 1990.

[11] K. Lange. An adaptive barrier method for convex programming. *Methods and Applications of Analysis*, 1(4):392–402, 1994.

[12] K. Lange. *Optimization.* Springer-Verlag New York, 2004.

[13] V. Ravi. Package “ alabama ”. `https://cran.r-project.org/web/packages/alabama/alabama.pdf`, 2015.

[14] K. Tanaka. Forecasting scenarios from the perspective of a reverse stress test using second-order cone programming. *Journal of Risk Model Validation*, 11(2):1–21, 2017.

[15] K. Tanaka. Forecasting conceivable interest rate market scenarios and significant losses on interest rate portfolios using mathematical optimization. `http://www.optimization-online.org/DB_FILE/2019/01/7024.pdf`, 2019.

[16] O. Tsuchiya. *Theory and practice of valuation adjustment(in Japanene).* Kinzai institute for financial affairs, inc, 2016.

[17] Y. Yajima and H. Konno. Outer approximation method for lower rank bilinear programming problems. *Journal of the Operations Research Society of Japan*, 38(2):230–239, 1995.