Calibration of financial models using quasi-Monte Carlo

Andreas Eichler and Gunther Leobacher and Heidrun Zellinger
Summer 2010

In the area of financial mathematics Monte Carlo simulation is often successfully used to estimate the prices of certain products. However in many cases calibrating Monte Carlo based models to market prices turns out to be difficult due to stochastic noise arising in the objective functionals.

This noise can be reduced by the use of fixed point-sets of random numbers which are reused for every new set of parameters (i.e. in every new step of the optimization algorithm used for calibration).

In this paper we argue that the above technique can be enhanced by using fixed low discrepancy point-sets (quasi-Monte Carlo method) instead of ones originating from Pseudo-Random-Number generators. The method is applied to two different financial models and the results are compared with the classical one.

Keywords: Calibration, parameter estimation, quasi-Monte Carlo, Monte Carlo, optimization

1 Introduction

In many cases, arbitrage-free prices of financial derivatives can be expressed as integrals over the $M$-dimensional unit cube, so that they can be computed using numerical integration schemes. When $M$ becomes high, the method of choice usually is Monte Carlo integration (MC), which is largely immune to the curse of dimension, i.e. the number of function evaluations needed to achieve a given degree of accuracy does not grow exponentially with the dimension $M$.

Given a function $f$ on the $M$-dimensional unit cube, one draws a finite sequence of random points $(x_1, \ldots, x_N) \in [0, 1]^M$ and estimates

$$
\int_{[0,1]^M} f(x) \, dx \approx \frac{1}{N} \sum_{n=1}^{N} f(x_n)
$$

The first author is supported by the Austrian Science Fund under Project Nr. P21196. The second author is partly supported by the Austrian Science Fund under Project Nr. P21196. The third author is supported by a DocF-Forte Grant of the Austrian Academy of Sciences.
In practice one usually generates pseudo-random numbers instead of true random numbers, i.e. deterministic sequences with statistical properties that make it resemble random numbers. Obviously, it may well happen that a particular randomly chosen set of points leads, by chance, to a very poor (or very good) approximation. For example, with positive probability all points lie in the same half of the cube. Fishman [15] provides a standard reference for Monte Carlo methods. Glasserman [18] provides a comprehensive treatment of Monte Carlo methods in finance.

A competing method for high dimensional integration is simulation using low-discrepancy points, also known as quasi-Monte Carlo (QMC). The philosophy behind that approach is that, when you already use uniformly distributed points to approximate an integral, then you might as well use non-random points which are constructed in order to approximate uniform distribution as fast as possible. There are many constructions known for such point sets which aim to minimize different notions of discrepancy, i.e. measures for the departure from uniform distribution. See for example [23] of [32] for different notions and estimates and constructions. See also [33] for a recent survey on low-discrepancy simulation.

If Monte Carlo has to compete against quasi-Monte Carlo, then the latter method – if properly applied – often gives better results in terms of accuracy and computation speed than the former. Particularly often this seems to be true for integration problems in finance, as was noted in [1, 34, 36, 35, 30, 21] among others.

Nevertheless, many practitioners as well as researchers prefer Monte Carlo methods to do their numerics, which is mainly due to the following reason: Monte Carlo methods may be applied without reformulating the problem in question as an integral over the unit cube. Besides being more straightforward to implement, this also has the advantage that one does not have to even think about the dimension of the underlying integration problem – it may even be infinite – and that one can use rejection algorithms for the generation of random variables. The second point is of particular interest, since rejection algorithms are usually fast and give (pseudo-)random points with the exact required distribution. On the other hand, the inversion method for generating random variables with a given distribution is usually rather slow and when the inverse is only approximated in order to gain speed, then the result may be biased.

So, while quasi-Monte Carlo bears the potential of faster estimation of integrals by using less function evaluations, Monte Carlo also scores some points here and it is not always clear which method gives the better overall performance. In finance, the picture changes if one does not talk about pricing, but about calibration. Calibration is an intrinsically important process in finance because it is usually the only way to get hold of the pricing measure. That means that, while some parameters may be estimated from price processes by statistical methods, other parameters have to be calibrated to market prices, i.e. the parameters have to be chosen in a way so that the model fits the observed prices best.

While usually not stating this explicitly, most researchers and practitioners do not use Monte Carlo for calibration. We will explain in Section 3 what (quasi-)Monte Carlo Calibration looks like and why true Monte Carlo usually is not the method of choice.

We give a brief summary of the remainder of the paper. Section 2 gives a short introduction to Monte Carlo and quasi-Monte Carlo integration. As already mentioned, in Section 3 we give good theoretical reasons for why QMC is preferred to MC for calibration. Section 4 gives an error estimate on the optimal model parameters for QMC calibration in a very special case. Sections 5 and 6 provide numerical examples which underline the theoretical reasoning. Section
(Quasi-)Monte Carlo integration

(Quasi-)Monte Carlo integration is a method to compute (high dimensional) integrals approximately by using randomly chosen resp. deterministic point sets. In the following we restrict to the integration of functions \( f \in L^1([0,1]^s) \) over the \( s \)-dimensional unit cube.

Suppose our objective is to calculate

\[
\alpha := \int_{[0,1]^s} f(x) \, dx \tag{1}
\]

which can be represented as

\[
\alpha = \mathbb{E}[f(X)] \tag{2}
\]

with \( X = (X_1, \ldots, X_s) \) where the \( X_i \sim \mathcal{U}[0,1] \) are i.i.d. for \( i = 1, \ldots, s \).

The pricing of financial products often involves the calculation of such expectations. For example \( f \) may be the result of transformations that converts the \( X_i \) to normal random variables, these to paths of underlying assets and the paths to a discounted payoff of a derivative security in a risk neutral measure. Therefore the numerical integration of \( f \) is important for many financial applications, see for example [25].

(Quasi-)Monte Carlo integration approximates the integral (1) by

\[
\hat{\alpha}_N := \frac{1}{N} \sum_{i=1}^{N} f(x_i) \tag{3}
\]

where the \( x_i \in [0,1]^s \) are randomly chosen (MC) resp. deterministically chosen (QMC) points in the hypercube \([0,1]^s\).

Note that \( f \) need not be available in an explicit form, we only need some way to evaluate \( f \). In ordinary MC-simulation we can produce a sequence of points of the \( s \)-dimensional unit cube by taking a scalar sequence \( x_1, x_2, \ldots \) of independent \( \mathcal{U}[0,1] \)-distributed random numbers and forming vectors \( x_1 = (x_1, \ldots, x_s) \), \( x_2 = (x_{s+1}, \ldots, x_{2s}) \), \ldots .

In QMC-integration the construction of the points \( x_i \) depends explicitly on the dimension, that means \( s \) must be known in advance (which makes for example acceptance-rejection methods inapplicable).

For the Monte Carlo integration the strong law of large numbers ensures that for integrable functions \( f \)

\[
\hat{\alpha}_N \to \alpha \quad \text{with probability 1 as } N \to \infty \tag{4}
\]

As the points \( x_i \) are randomly chosen we only obtain probabilistic error estimates for the integration error \(|\alpha - \hat{\alpha}_N|\). It is easy to prove that

\[
\mathbb{E}[|\alpha - \hat{\alpha}_N|] \leq \frac{\sigma_f}{\sqrt{N}} \tag{5}
\]
where $\sigma_f$ is the standard deviation of $f$, which in the case of an $L^2$-function $f$ is given by

\[
\sigma_f = \left( \int_{[0,1]^s} (f(x) - \alpha)^2 \, dx \right)^{1/2}
\]

Note that the expected integration error is independent of the dimension $s$ in the case of MC-integration.

In contrast to MC deterministic error bounds are available for Quasi-Monte Carlo integration. Here the classical key result is the Koksma-Hlawka inequality, which states that for any $x_1, \ldots, x_N \in [0,1]^s$

\[
|\alpha - \hat{\alpha}_N| \leq V(f) \cdot D_N^s
\]

where $V(f)$ is the variation of $f$ in the sense of Hardy and Krause and $D_N^s$ is the star discrepancy of the point set $\{x_1, \ldots, x_N\}$. The discrepancy measures the deviation of the point set from uniformity. In the case of Monte Carlo integration we have seen that the (probabilistic) error bound is of the form $O(N^{-1/2})$. What do we know about the magnitude of the quasi-Monte Carlo error bound in (7) resp. about error bounds for $D_N^s$?

By explicit construction we know that for arbitrary $s$ there exist $s$-dimensional sequences $x_1, x_2, \ldots$ with

\[
D_N^s \leq c \cdot \frac{(\log N)^s}{N} \quad \text{for all } N,
\]

where $c$ only depends on $s$. Examples are the Halton, Faure, Sobol or Niederreiter sequence (see [27]).

Asymptotically the logarithmic term becomes negligible, so quasi-Monte Carlo methods provide error bounds $O\left(N^{-(1-\epsilon)}\right)$, for all $\epsilon > 0$, which theoretically is by far superior to the Monte Carlo error bound $O\left(N^{-1/2}\right)$. In recent years new methods have been developed which are able to take into account higher regularity of the function $f$, giving faster convergence rates for more regular $f$. For example it is shown by Dick in [12] that if for $f : [0,1]^s \to \mathbb{R}$ the mixed partial derivatives up to order $\alpha \geq 1$ exist and are square integrable, then there is an integration rule which has convergence of order $O\left(N^{-(\alpha-\epsilon)}\right)$ for all $\epsilon > 0$.

However for values of $N$ used in practice the asymptotic boundary may not be relevant, especially for high dimensions $s$ one may expect that Monte Carlo methods yield better results than quasi-Monte Carlo simulation.

Especially in financial applications the dimension is often very high. The two main sources of high dimensions are the number of underlying assets (as in the calibration of CDO tranches in Section 5) or the number of time steps when simulating paths of underlying dynamics (as in the Heston-model given in Section 6). Therefore at first sight it seems unfavourable to use quasi-Monte Carlo methods for these problems. But many recent applications in finance have shown that QMC methods are effective even in high dimensions, which may lie in the structure of problems arising in finance, because often the high-dimensional integrals may be rewritten or approximated by much lower-dimensional integrals. Sloan and Wóźniakowski [39] have identified classes for which QMC methods are effective in high dimensions because of the decreasing importance of higher dimensions. An application of these ideas to the Brownian bridge construction is given in Larcher, Leobacher and Scheicher [26].
The above cautious remarks notwithstanding it has been demonstrated in a number of articles, see e.g. [36], that many functions originating from finance exhibit fast convergence of QMC-integration even if the formal requirements, i.e. low dimensionality and smoothness (resp. boundedness of variation) are not satisfied. Also the function occurring in our example in Sections 5 does not meet the requirement for smoothness, though the irregularities are not particularly malign. It turns out, however, that convergence of the QMC algorithm using Sobol points is faster than for MC, see Eichler et al. [14] for details.

3 (Quasi-)Monte Carlo Calibration

What does simulation-based calibration look like? Suppose we have a number of financial derivatives with observed prices \( P_{\text{obs}}^1, \ldots, P_{\text{obs}}^I \) and we want to find the set \( \theta \in \Theta \) for which the model replicates those prices most accurately. For a fixed set of parameters \( \theta \) one generates a set of random numbers \( \{x_1, \ldots, x_N\} \) with \( x_n \in \mathbb{R}^M \) which are used for pricing the \( I \) derivatives simultaneously. The Monte Carlo estimators for the prices of the \( I \) products are given by

\[
\hat{P}_i(\theta) = \frac{1}{N} \sum_{n=1}^{N} (\text{Payoff}_i(x_n, \theta)) \quad i = 1, \ldots, I
\]

The deviation from the observed market prices is then measured by some error functional depending on \( \hat{P}_1(\theta), \ldots, \hat{P}_I(\theta) \) and \( P_{\text{obs}}^1, \ldots, P_{\text{obs}}^I \), for example the quadratic error functional

\[
\text{Err}(\theta) = \sum_{i=1}^{I} (\hat{P}_i(\theta) - P_{\text{obs}}^i)^2
\]

We are looking for some optimal parameter set \( \theta^* \), i.e. \( \text{Err}(\theta^*) = \min_{\theta \in \Theta} \text{Err}(\theta) \).

Note that application of the Monte Carlo method means using a different random point set \( \{x_1, \ldots, x_N\} \) for every new set of model parameters \( \theta \). That means in particular, that \( \text{Err}(\theta) \) for a given \( \theta \) is a random variable.

This has a huge practical downside: the error functional becomes distorted by random noise, like in Figure 1. This makes it very hard to find an optimal set of parameters, though algorithms exist for this purpose, see [6]. However those stochastic optimization techniques are rather time consuming.

As a consequence, usually one fixed point set \( \{x_1, \ldots, x_N\} \) is used for all \( \theta \)'s for which the error functional has to be evaluated, see Figure 2.

In practice this means that the random number generator is started afresh before the error functional is evaluated for a new set of parameters. This method has been termed “method of dependent sampling” in the Monte Carlo community, see for example Kurbanmuradov and Sabelfeld [24].

This “recycling” of the same point set actually makes the Monte Carlo pricing used in the calibration look more like quasi-Monte Carlo, where the deterministic point set is generated at random. But, if one is going to use the same point set for all evaluations, why would one generate
Figure 1: Error functional $Err(\theta)$ of the market and models spreads of the CDX.NA.IG tranches for a one-factor Lévy gamma model with parameters $\theta = (\gamma, \phi)^\top$ in a Monte Carlo setting with $N = 5000$ simulations.

Figure 2: Error functional $Err(\theta)$ of the market and models spreads of the CDX.NA.IG tranches for a fixed point set with $N = 5000$ simulations.
this point set at random instead of using a point set with known good distribution properties, such as low-discrepancy sequences?

One might be tempted to answer this question in the same way as earlier, when we discussed the reasons for the popularity of the Monte Carlo method for pricing. But there is an important difference: instead of a single value, we have to compute several values of the error functional, which we want to use to minimize this functional subsequently. But the set of elements of the pseudo-random sequence that is discarded by the rejection algorithm usually depends strongly on the parameters, thereby making the scenarios depend on the parameters in a discontinuous fashion. This adds some non-random noise to the estimator of the error-functional.

Example 1. Let \((X^\lambda_i)_{i=1,...,n}\) be a sequence of i.i.d. Gamma(\(\lambda, 1\)) random variables and \(S^\lambda = \sum_{i=1}^{n} X^\lambda_i\). Let further \(f(s) := s - \bar{\lambda} \cdot n\).

We want to approximate \(\alpha(\lambda) = \mathbb{E}[f(S^\lambda)]\) by the estimator

\[\hat{\alpha}_N(\lambda) = \frac{1}{N} \sum_{j=1}^{N} f(S^\lambda_j)\]

for different \(\lambda \in (\bar{\lambda} - \varepsilon, \bar{\lambda} + \varepsilon)\)

- using a Monte Carlo method based on acceptance-rejection with exponential distribution as dominating function where the pseudo number generator is reset each time the parameter \(\lambda\) changes
- using a low discrepancy quasi-Monte Carlo sequence (e.g. a Sobol sequence) and the inverse transform method

We draw those functions for \(n = 5, N = 1024, \bar{\lambda} = 2\) and \(\varepsilon = 0.2\), where \(\lambda\) changes in steps of 0.001. In Figure 3 one can see a rather strong noise as compared to Figure 4. A fixed random point set would also reduce the noise analogous to Figure 4, when combined with the inverse transform method.

The same effect will usually occur for problems of infinite dimension.

Example 2. Consider a similar problem as in Example 1, but with

\[S^\lambda = \sum_{i=1}^{n} U_i,\]

where \(X^\lambda\) is a Poisson process with intensity \(\lambda\) and \(U_1, U_2, \ldots\) are independent uniform on [0, 1].

It is quite obvious that small changes of \(\lambda\) may have a relatively big impact on \(\frac{1}{N} \sum_{j=1}^{N} f(S^\lambda_j)\), since even such a small change may change \(X^\lambda_i\) by one. But then all succeeding pseudo-random numbers are mapped to other random variables of the problem than before, thereby changing the outcome arbitrarily.
A recent paper illustrating how to deal\(^1\) with infinite-dimensional problems with quasi-Monte Carlo is [7].

At this point we want to reiterate the point made earlier: if we have to use a deterministic point set when calibrating and if we have to take extra care in the simulation anyway, it is reasonable to take point sets with known good distribution properties. We have done exactly this with two models from finance, and the numerical results indicate that model calibration can benefit from quasi-Monte Carlo methods in many different settings.

In Section 5 we describe a one factor Lévy-structural model as proposed in [8] and [3] for the

\(^1\)Actually the cited paper does only deal with pricing. Calibration may need some extra effort.
valuation of CDOs. We calibrated the parameters of this credit risk model to hypothetical and real market tranche spreads of CDOs.

The second model that we investigated in Section 6 is the well-known Heston stochastic volatility model [19] which we calibrated to a set of 100 European call options. Therefore we used an adjoint method as proposed in [22]. It is worth noting that in this model there is no need to take special care in order to guarantee that the same pseudo- or quasi-random numbers are always mapped to the same random variable in the (discretized) model. This is due to the nature of the generation of random normal variables with expectation $\mu$ and variance $\sigma^2$: first a standard normal variable is generated, which is subsequently multiplied by $\sigma$ and increased by $\mu$. As a consequence, even if a rejection method (e.g. Marsaglia-Bray, [29]) is used, there is no noise since for each choice of $\mu, \sigma$, always the same set of pseudo-random numbers is discarded. This is probably the reason why many practitioners and researchers rarely encounter the problems described earlier when calibrating their models: most models solely depend on the generation of standard normals. Nevertheless, in order to get a smooth error functional, one has to “recycle” the pseudo-random numbers (by resetting the generator or storing the numbers) and so it still seems more reasonable to prefer point sets with known good distribution properties over a random point set.

For both financial models we give numerical results in which we compare the performance of deterministic point sets generated by pseudo-random numbers (in the following we speak of fixed Monte Carlo point sets) and quasi-Monte Carlo point sets (generated by a low-discrepancy sequence) in simulation-based parameter calibration.

We want to end this section with a word of caution: if the functions being integrated is very irregular and the effective dimension is high, Monte Carlo might be the better approach in that it provides us with an error bound, if only a probabilistic one. Kurbamuradov and Sabelfeld [24] give an error bound for the dependent sampling method that might be useful for calibration, though probably not using an error estimate similar to the one presented in the next section.

4 Error estimate for quadratic error functionals

In this section we will give an estimate on the rate of convergence of calibrated model parameters to the true ones for a special form of the error functional.

While the error functionals in our problems as described in sections 5 & 6 are not of this special form, the estimate given here still provide an indication about what order of convergence one might expect. This point will be discussed in more detail in Subsection 4.2

4.1 Problem formulation

We are given data $p_1, \ldots, p_m$ for which a theory predicts that they are of the form $p_k = F_k(\theta)$. We want to find the parameter $\theta^* \in \mathbb{R}^n$ which fits the data best in that the functional

$$\phi(\theta) = \sum_{k=1}^{m} (F_k(\theta) - p_k)^2$$
is minimized in $\theta = \theta^*$. Suppose the $F_k$ are intrinsically hard to calculate because they contain, for example, very high-dimensional integrals.

We assume that we have some method for approximating the $F_k$, that is there are functions $G_1, \ldots, G_m$ such that on some domain containing $\theta^*$ we have that $G_k$ is close to $F_k$. Now we can try to minimize

$$
\psi(\theta) = \sum_{k=1}^{m} (G_k(\theta) - p_k)^2
$$

instead of $\phi(\theta)$. Suppose $\tilde{\theta}$ is such that for some $\varepsilon > 0$

$$
\psi(\tilde{\theta}) < \inf_{\theta} \psi(\theta) + \varepsilon.
$$

How far can $\tilde{\theta}$ be from $\theta^*$?

### 4.2 Assumptions

In order to find some estimates we make a number of (strong) assumptions on the $F$’s and $G$’s.

- For all $k$ we have that $F_k$ is linear in $\theta$ on some environment of the optimal parameter set, such that we can write $F_k(\theta) - p_k = a_k \theta + b_k$;

- For all $k$ we have that $G_k$ is almost linear in $\theta$ in the sense that we can write $G_k(\theta) - p_k = a_k \theta + b_k + h_k(\theta)$, where $|h_k(\theta)| < \gamma$ and where $\gamma$ is a constant that can be made small.

Note that the section assumption essentially states that there exists a deterministic error bound of the size $\gamma$.

We write $a_k = (a_{k,1}, \ldots, a_{k,n})$ and we denote by $A$ the matrix composed of the row vectors $a_1, \ldots, a_m.$ Further we write $b$ for the column vector $(b_1, \ldots, b_m)^\top$ and $h(\theta) = (h_1(\theta), \ldots, h_m(\theta))^\top$.

With this notation we have $\phi(\theta) = \|A \theta + b\|^2$, where $\|\cdot\|$ denotes the usual Euclidean norm on $\mathbb{R}^m$.

At first glance, the above assertion look rather innocuous. Let us stress however, that we assert that $F_k$ is linear, not only that we can approximate it with arbitrary precision by a linear function on a small subdomain.

Nevertheless it is not inconceivable that the estimate given below has some significance to practical problems. Figures 5 & 6 illustrate this. Figure 5 shows a two-dimensional projection of $F(\theta) - p$ for the problem treated in detail in Section 5. Obviously, $F(\theta) - p$ is not a line for this problem, but in the magnified picture, Figure 6, we see that it is very close to its tangent. This is due to the fact that $F(\theta) - p$ is smooth and that $F(\theta^*) - p$ is very close to 0.

### 4.3 Error and complexity

**Proposition 1.** For all $\theta \in \mathbb{R}^n$ we have $|\psi(\theta)^{1/2} - \phi(\theta)^{1/2}| \leq \gamma m^{1/2}$.
Figure 5: A projection of $F(\theta) - p$ for the CDO-calibration problem

**Proof.** We have
\[
\psi(\theta) = \|A\theta + b + h(\theta)\|^2 \\
\leq \left(\|A\theta + b\|^2 + 2\|A\theta + b\|\|h(\theta)\| + \|h(\theta)\|^2\right) \\
\leq \left(\|A\theta + b\|^2 + 2\|A\theta + b\|\gamma m^{1/2} + \gamma^2 m\right) \\
= \left(\phi(\theta)^{1/2} + \gamma m^{1/2}\right)^2,
\]
which yields one part of the inequality.

\[
\phi(\theta) = \|A\theta + b + h(\theta) - h(\theta)\|^2 \\
\leq \|A\theta + b + h(\theta)\|^2 + 2\|A\theta + b + h(\theta)\|\|h(\theta)\| + \|h(\theta)\|^2 \\
\leq \left(\psi(\theta)^{1/2} + \gamma m^{1/2}\right)^2,
\]
which gives the other inequality. \(\square\)

**Corollary 1.** Let $\phi(\theta^*) = \inf_\theta \phi(\theta)$ and $\psi(\hat{\theta}) = \inf_\theta \psi(\theta)$. Then $|\psi(\hat{\theta})^{1/2} - \phi(\theta^*)^{1/2}| \leq \gamma m^{1/2}$.\(\square\)

**Proof.** $\psi(\hat{\theta})^{1/2} \leq \psi(\theta^*)^{1/2} \leq \phi(\theta^*)^{1/2} + \gamma m^{1/2}$, and similar for the other inequality. \(\square\)

Let $\phi(\theta^*) = \inf_\theta \phi(\theta)$, i.e. let $\theta^*$ be a minimizer of $\sum_{k=1}^m (a_k \theta + b_k)^2$. It is readily verified (see [9]) that $\theta^*$ is unique iff $A^\top A$ is invertible and in that case
\[
\theta^* = - (A^\top A)^{-1} A^\top b.
\]
In the general case where $A^TA$ may fail to be invertible we note that if $\|A\theta^* + b\|^2 = \inf_\theta \|A\theta + b\|^2$, then for all $\theta$

$$\|A\theta^* + b\|^2 \leq \|A\theta + b\|^2$$

$$= \|A(\theta - \theta^*) + A\theta^* + b\|^2$$

$$= \|A(\theta - \theta^*)\|^2 + 2\langle A(\theta - \theta^*), A\theta^* + b \rangle + \|A\theta^* + b\|^2$$

i.e. $\|A(\theta - \theta^*)\|^2 + 2\langle A(\theta - \theta^*), A\theta^* + b \rangle \geq 0$ for all $\theta$. But from this it follows that $\langle A(\theta - \theta^*), A\theta^* + b \rangle = 0$ for all $\theta$ (if $2\langle Ax, y \rangle + \langle Ax, Ax \rangle \geq 0$ for all $x$ then for all $\lambda < 0$ we have $2\lambda \langle Ax, y \rangle + \lambda^2 \langle Ax, Ax \rangle = 2\langle A(\lambda x), y \rangle + \langle A(\lambda x), A(\lambda x) \rangle \geq 0$, such that $2\langle Ax, y \rangle \leq (-\lambda)\|Ax\|^2$ and therefore $\langle Ax, y \rangle \leq 0$. Similarly $\langle Ax, y \rangle \geq 0$). For well known orthogonality results for least squares problems see [9].

As a consequence,

$$\|A\theta + b\|^2 = \|A(\theta - \theta^*)\|^2 + \|A\theta^* + b\|^2$$

for all $\theta$ or, in other words,

$$\phi(\theta) = \|A(\theta - \theta^*)\|^2 + \phi(\theta^*). \quad (11)$$

**Proposition 2.** Let $\hat{\psi}(\hat{\theta}) = \inf_\theta \psi(\theta)$. (Note that $\psi$ and therefore $\hat{\theta}$ depend on $\gamma$.)

Then there exists $C > 0$ depending only on $A$ and $b$ such that

$$\|A(\hat{\theta} - \theta^*)\| \leq C\sqrt{\gamma}.$$
In the case where $A^\top A$ is invertible there exists $C_1 > 0$ depending only on $A$ and $b$ such that

$$\|\hat{\theta} - \theta^*\| \leq C_1 \sqrt{\gamma}.$$  

**Proof.** We have

$$\phi(\hat{\theta})^{1/2} \leq \psi(\hat{\theta})^{1/2} + \gamma m^{1/2} \leq \psi(\theta^*)^{1/2} + \gamma m^{1/2} \leq \phi(\theta^*)^{1/2} + 2\gamma m^{1/2},$$

such that with equation (11)

$$\|A(\hat{\theta} - \theta^*)\|^2 = \phi(\hat{\theta}) - \phi(\theta^*)$$

$$\leq (\phi(\theta^*)^{1/2} + 2\gamma m^{1/2})^2 - \phi(\theta^*)$$

$$\leq 4\phi(\theta^*)^{1/2}\gamma m^{1/2} + 4\gamma^2 m$$

$$\leq 5\phi(\theta^*)^{1/2}\gamma m^{1/2}$$

for $\gamma$ small enough. Therefore $\|A(\hat{\theta} - \theta^*)\| \leq C_1 \sqrt{\gamma}$. In the case where $A^\top A$ is invertible, we have uniqueness of $\theta^*$ and $\|\hat{\theta} - \theta^*\| \leq C_1 \sqrt{\gamma}$. 

If $\tilde{\theta}$ is only an approximate minimizer of $\psi$ such that $\psi(\tilde{\theta}) \leq \psi(\hat{\theta}) + \varepsilon$, then we may still argue that

$$\phi(\tilde{\theta})^{1/2} \leq \psi(\hat{\theta})^{1/2} + \gamma m^{1/2} \leq \psi(\hat{\theta})^{1/2} + \varepsilon + \gamma m^{1/2} \leq \phi(\theta^*)^{1/2} + \varepsilon + 2\gamma m^{1/2},$$

such that

$$\|A(\tilde{\theta} - \theta^*)\| \leq C_2 \sqrt{\varepsilon + 2\gamma m^{1/2}}.$$ 

In the case where $A^\top A$ is invertible we have

$$\|(\tilde{\theta} - \theta^*)\| \leq C_3 \sqrt{\varepsilon + 2\gamma m^{1/2}}.$$ 

Note that all the constants only depend on the calibration problem, that is on $A$ and $b$.

We now turn our attention to the complexity of the calibration: Assume that we want to find a minimizer $\hat{\theta}$, such that we want to reach an accuracy level $\delta$:

$$\|A(\hat{\theta} - \theta^*)\| \leq C_2 \sqrt{\varepsilon + 2\gamma m^{1/2}} \leq \delta$$  \hspace{1cm} (12)

We further assume that halving $\gamma$ resp. $\varepsilon$ results in a $2^\beta$ resp. $2^\alpha$-factor in computational effort, where $\alpha, \beta > 0$. For the computational time cost-function in our optimization algorithm then one can write

$$\text{cost}(\varepsilon, \gamma) \propto \frac{1}{\varepsilon^\alpha} \cdot \frac{1}{\gamma^\beta}$$

In order to minimize our computational effort we have to find an optimal $\gamma$ and $\varepsilon$ to reach at least accuracy $\delta$. Therefore we have to solve the optimization problem

$$\text{cost}(\varepsilon, \gamma) \rightarrow \min$$

with constraint $\varepsilon + 2\gamma m^{1/2} = \left(\frac{\delta}{C_2}\right)^2$. As a result

$$\text{cost}(\delta) \propto \frac{1}{\delta^{2(\alpha+\beta)}}$$  \hspace{1cm} (13)
5 Calibration of a credit risk model

We tested the performance of fixed Monte Carlo point sets versus quasi-Monte Carlo point sets for the calibration of a credit risk model, i.e. the one-factor Lévy-Gamma model proposed in [8] or [3]. In Section 5.1 we will describe the main ideas of the valuation of tranches of Collateralized Debt Obligations (CDOs) and widely-used simplifying assumptions of this model (for more details see [14]). Numerical results of the calibration are given in Section 5.2.

5.1 Basics on valuation of CDO tranches

Collateralized Debt Obligations (CDOs) are one of the most important portfolio credit derivatives. They cover the risk of defaults in the underlying portfolio which consists of credit risky bonds or – in the case of synthetic CDOs – of Credit Default Swaps (CDS). Generally, CDOs are divided into several tranches, which split-up the overall risk of default.

When trading CDO tranches the so called Protection Seller gains periodic payments via a “tranche spread” from the Protection Buyer, until maturity $T$ or the time of default, in exchange for covering losses in the tranche. By valuation of a CDO tranche we mean determining the fair tranche spread $s^i$ of the $i$-th tranche with respect to the risk neutral measure. We obtain $s^i$ from the equation

$$PL_i (s^i) = DL_i$$

where $PL_i$ denotes the expected premium payments (Premium Leg) and $DL_i$ the expected default payments (Default Leg) of the $i$-th tranche. In order to model $PL_i$ and $DL_i$ one has to model the loss of the portfolio. Let $(N_t)_{t=0}^T$ denote the nominal process of the portfolio and $(L_t)_{t=0}^T$ its loss process. Without loss of generality we assume $N_0 = 1$ then $L_t = 1 - N_t$. The $i$-th tranche covers a fixed percentage $[K_{i-1}, K_i) \subset [0, 1]$ of the whole portfolio loss, therefore one easily obtains the loss $(L_t^i)_{t=0}^T$ and nominal process $(N_t^i)_{t=0}^T$ of the $i$-th tranche from the processes $L_t$ resp. $N_t$.

$$L_t^i = \min (K_i - K_{i-1}, \max (L_t - K_{i-1}, 0))$$

(15)

$$N_t^i = \max (0, \min (K_i - K_{i-1}, N_t - (1 - K_i)))$$

(16)

In terms of these two processes we can write the expected Default and Premium Leg as

$$DL_i = \mathbb{E} \left( \int_0^T e^{-rt} dL_t^i \right)$$

(17)

and

$$PL_i = \mathbb{E} \left( s^i \cdot \int_0^T e^{-rt} N_t^i dt \right)$$

(18)

respectively. Using Equations (14), (17) and (18) we therefore obtain for the fair spread

$$s^i = \frac{\mathbb{E} \left( \int_0^T e^{-rt} dL_t^i \right)}{\mathbb{E} \left( \int_0^T e^{-rt} N_t^i dt \right)}$$

(19)

Once we have a model for the nominal loss process we can compute or give estimators for the expectations in (19). In recent years, several models have been proposed, which can be divided
into two main classes: default-intensity approaches (see for example [13], [16], [37] or [38]) and structural models (one of the first was the one-factor Gauss copula model proposed by [28], which has been extended and modified e.g. by [5] or [20]). For our numerical examples we will use a model of the second class, i.e. a one-factor Lévy structural model.

For simplicity we consider a fixed interest rate $r$ and assume that the portfolio consists of $J$ equally weighted products – which is fulfilled for many CDOs on the market – with fixed recovery rate $R \in [0, 1]$. That means each time a product defaults the occurred loss is $(1 - R)/J$.

Let $\tau_j$ denote the random time at which the $j$-th credit defaults, then the loss process is given by

$$L_t = \sum_{j=1}^{J} \frac{1 - R}{J} \cdot 1_{[\tau_j, \infty)}(t)$$

In the one-factor Lévy structural model proposed in [8] and [3] the default time $\tau_j$ is modelled by

$$\tau_j = \inf\{t \geq 0 : V^j_t \leq b_j\}$$

with

$$V^j_t = V^j_0 \cdot \exp\left(-X^j_t + \mu_j t\right)$$

where $X^j$ is a Lévy process and the drift factor $\mu_j$ is choosen such that the process $V^j$ is a martingale.

The idea behind this model is that $V^j$ reflects the underlying firm’s value of the $j$-th credit in the portfolio and default occurs as soon as the firm’s value hits the fixed barrier $b_j$.

$X^j$ is a Lévy process associated with the $j$-th product. We make a number of additional assumptions on the form of the $X^j$’s. Let $Y$ be a Lévy process with $0 < \mathbb{E}(e^{-Y_1}) < \infty$ and $\mathbb{E}(Y^2_1) < \infty$ and let $Y^g, Y^1, \ldots, Y^J$ be independent copies of $Y$. Then we assume

$$X^j_t := Y^g_{\phi t} + Y^j_{(1-\phi)t}$$

for some fixed $\phi \in (0, 1)$.

We interpret $Y^g$ as a global risk, which is the same for all products and $Y^j$ as the idiosyncratic risk of the $j$-th product. The parameter $\phi$ equals the correlation between these two processes.

Note that due to the special form of the processes we have $\mu_1 = \ldots = \mu_J =: \mu$.

In terms of the Lévy process $X^j$ the default time $\tau_j$ is given by

$$\tau_j = \inf\{t \geq 0 : X^j_t \geq b_j + \mu t\}$$

with fixed barriers $b_1, \ldots, b_J$.

For our numerical results (see Section 5.2) we modelled $X^1, \ldots, X^J$ as Gamma processes, in [8] and [3] also the following Lévy processes have been studied: Brownian motion, Variance gamma and Jump diffusion.

In our model we further assume that default can only appear at discrete times $0 = t_0 < t_1 < \cdots < t_K = T$ (which is known as Bermudian approximation), i.e. we approximate the default times $\tau_j$ in Equation (24) by

$$\tau_j = \min\{t_k : X^j_{t_k} \geq b_j + \mu t_k\}$$
The one-factor Lévy structural model described above allows us to compute the tranche spreads $s^i$ from Equation (19) approximately, for example by Monte Carlo simulation. However in order to obtain an analytical pricing formula more simplifying assumptions are needed. These are

**Assumption 1 (European approximation).** It is assumed that default can only occur at maturity $T$.

**Assumption 2 (Homogeneity of the portfolio).** It is assumed that all risks are identical, i.e. $b_j$ is the same for all $j = 1, \ldots, J$.

As we included these assumptions in our calibration we were able to compare the numerical results of the calibration with fixed Monte Carlo resp. quasi-Monte Carlo point sets with the analytical (semi-exact) pricing formula. Certainly these assumptions influence the estimated parameters of the calibration. Numerical experiments in [14] indicate that the error grows with the inhomogeneity of the portfolio and the increasing default risk (i.e. spreads) of the contained products.

It is worth noting that even under Assumptions 1 and 2 the integrand in Equation (17) does not depend on the random variables $Y^g, Y^1, \ldots, Y^J$ in a smooth way, so that the classical error estimates for QMC integration cannot be applied with rigour.

In the following numerical examples we calibrate the tranches of the CDX.NA.IG to the model. As this index consists of products with relatively low credit spreads the error caused by simplifying assumptions above is tolerable.

### 5.2 Calibration and numerical results

For calibration we regard the one-factor Lévy structural model described in Section 5.1 where $X^1, \ldots, X^J$ are Gamma processes, i.e. we have $Y^j_t \sim \text{Gamma}(\gamma \phi t, 1)$ and $Y^j_t \sim \text{Gamma}(\gamma (1-\phi) t, 1)$ for $j = 1, \ldots, J$. The drift parameter is given by $\mu = \gamma \cdot \log(2) - r$.

The model parameters are $\theta = (\gamma, \phi)^T$ with $\gamma > 0$ and $0 < \phi < 1$, the recovery rate $R$ and the barrier $b$. The two parameters $R$ and $b$ are not included in the parameter vector $\theta$, because $R$ is estimated from historical data and for given tranche spreads $b$ can be directly computed. Our aim is to calibrate this model to given tranche spreads $s^i_{\text{obs}}, i = 1, \ldots, I$ of the market. To that end we minimize the error functional

$$\text{Err}(\theta) = \sum_{i=2}^I (\hat{s}^i(\theta) - s^i_{\text{obs}})^2$$

Note that we neglect the first tranche in our error functional as proposed in [8] as the value of $s^1_{\text{obs}}$ is very high compared with the spreads of the other tranches. One could certainly use other weighting schemes to reduce the influence of the first tranche.

For calibration of the model we used the Assumptions 1 and 2 which allow us to compute $\hat{s}^i(\theta)$ either via simulation (with a fixed Monte Carlo or quasi-Monte Carlo point set) or with a semi-exact pricing formula.

At first we calibrated the model to tranches of the CDX.NA.IG with maturity $T = 5$ which
consists of $J = 125$ products. The observed market spreads are given in Table 1. In a second calibration we regarded a hypothetical, homogeneous portfolio consisting of $J = 125$ CDSs with a single spread of $s = 40$ bp each, which can only default at maturity $T = 5$. We priced the tranches of this CDO via Monte Carlo simulation with $N = 1000000$ runs for the parameters $\gamma = 0.4$ and $\phi = 0.1$ (the values of $s_{\text{obs}}^i$ are given in Table 1). In this way we can eliminate the error that originates from the Ass. 1 and 2 in our calibration.

For both cases we assume $r = 0.03$ and $R = 0.4$.

<table>
<thead>
<tr>
<th>Tranche $i$</th>
<th>covered loss</th>
<th>market spreads $s_{\text{obs}}^i$ of the 5y CDX.NA.IG in bp</th>
<th>market spreads $s_{\text{obs}}^i$ of a hypothetical, homogeneous tranched CDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0% - 3%</td>
<td>1169.58</td>
<td>1109.74</td>
</tr>
<tr>
<td>2</td>
<td>3% - 7%</td>
<td>103.50</td>
<td>96.13</td>
</tr>
<tr>
<td>3</td>
<td>7% - 10%</td>
<td>23.00</td>
<td>27.01</td>
</tr>
<tr>
<td>4</td>
<td>10% - 15%</td>
<td>11.00</td>
<td>14.06</td>
</tr>
<tr>
<td>5</td>
<td>15% - 30%</td>
<td>5.50</td>
<td>5.56</td>
</tr>
<tr>
<td>6</td>
<td>30% - 100%</td>
<td>2.54</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Table 1: Tranches and tranche spreads $s_{\text{obs}}^i$ used for calibration

In both calibrations we identified the unknown parameters $(\gamma, \phi)^T$ by minimizing the error functional in Equation (26) with a sequential quadratic programming algorithm with the bounds $0.05 \leq \gamma \leq 1$ and $0.01 \leq \phi \leq 0.99$. We compared the performance of 4 different fixed Monte Carlo point sets with a quasi-Monte Carlo point set generated by the Sobol sequence for $N = 2^{k+k}$, $k = 1, 2, \ldots, 6$ simulations.

When calibrating the CDX.NA.IG we used the optimal parameters $\theta^*_\text{se}$ from the semi-exact analytical pricing formula as reference value. We obtained from the minimization $\theta^*_\text{se} = (0.3793, 0.0729)^T$ and $\text{Err}(\theta^*_\text{se}) = 5.4245$.

In the second calibration the reference values are the given parameters $\theta_{\text{hyp}} = (0.4, 0.1)^T$.

Results of the calibration are given in the Figures 7 to 10 where the errors in the parameters $\gamma$ and $\phi$ are plotted.
Figure 7: Common logarithm of the relative error in $\gamma$ in dependence of the number of simulations $N$ for the calibration to the CDX.NA.IG.

Figure 8: Common logarithm of the relative error in $\phi$ in dependence of the number of simulations $N$ for the calibration to the CDX.NA.IG.
Figure 9: Common logarithm of the relative error in $\gamma$ in dependence of the number of simulations $N$ for the calibration to the hypothetical market.

Figure 10: Common logarithm of the relative error in $\phi$ in dependence of the number of simulations $N$ for the calibration to the hypothetical market.
The figures show that performance of the calibration is quite good for the Sobol point set - the relative error between the simulated and the optimal parameter is in most of the cases less than 10\%, for $N > 10^{13}$ runs it is always less than 5\%. The fixed Monte Carlo point sets sometimes provide a smaller error than the quasi-Monte Carlo point set, but in some cases the fit is quite bad (especially in the parameter $\gamma$).

Note that a large relative error in the parameters does not automatically mean that the minimum of the error functional $\text{Err}(\theta^*)$ is also large. For example the calibration to the CDX.NA.IG with the first fixed Monte Carlo point set we obtained for $N = 2^{11}$ the $\text{Err}(\theta^*) = 6.6$ which is comparable to $\text{Err}(\theta^*_\text{st})$, but the estimated parameters are $\theta^* = (0.7359, 0.0329)^\top$ which means a relative deviation of 90\% resp. 50\% compared to the parameters $\theta^*_\text{st}$.

For a fixed Monte Carlo point set there is no guarantee that the point set has good distribution properties. This is especially a problem when calibrating more complex models, where no analytical pricing formulas are available, e.g. the described model without the Assumptions 1 and 2, so there are no reference values for the optimal parameters any more.

6 Calibration of a stochastic volatility model

Another typical example for calibrating a financial market model to the prices of liquidly traded products is the calibration of stock price models to call option prices. Here we want to calibrate the well-known Heston stochastic volatility model [19] to the prices of $I^2$ European call options with different strikes $K_i$ and maturities $T_i$, $i = 1, \ldots, I$. In Section 6.1 we describe the model and the method we used for calibration. In Section 6.2 we give numerical results for calibrations with fixed Monte Carlo resp. quasi-Monte Carlo point sets.

6.1 The Heston Model and its calibration

The Black-Scholes model [10] is still widely used in applications for pricing options of different types. However the Black-Scholes model has a number of disadvantages, one of them being that it cannot be calibrated to European call options with different strikes and maturities, because of the strong assumption that the stock returns are normally distributed with fixed volatility. However a calibration is possible if one allows the volatility $\nu_t$ to be stochastic as it is done in the so-called Heston model. In that model the stock price $S_t$ (scaled such that $S_0 = 1$) and the scaled volatility $\tilde{\nu}_t = \nu_t/\nu_0$ are, under the pricing measure, driven by the SDE’s

\begin{align}
\frac{dS_t}{S_t} &= (r - \delta)dt + \sqrt{\nu_0 \tilde{\nu}_t} dW_t^1, \quad S_0 = 1, \\
\frac{d\tilde{\nu}_t}{\nu_0} &= \kappa \left( \frac{\theta}{\nu_0} - \tilde{\nu}_t \right) dt + \frac{\sigma}{\sqrt{\nu_0}} \sqrt{\tilde{\nu}_t} \left( \rho dW_t^1 - \sqrt{1 - \rho^2} dW_t^2 \right), \quad \tilde{\nu}_0 = 1
\end{align}

Here $x^+ := \max(x, 0)$ for any real number $x$, as usual. The parameters of the model are $\theta = (\nu_0, \kappa, \theta, \sigma, \rho)^\top$, where $\nu_0$ denotes the initial variance, $\theta$ the mean-reversion level of $\nu_t$, $\kappa$ the mean reversion speed, $\sigma$ the volatility of the variance and $\rho \in (-1, 1)$ a suitable correlation coefficient. $W_t^1$ and $W_t^2$ are independent Brownian motions. The interest rate $r$ and the dividend yield $\delta$ are assumed to be given constants.
With $Y_t := (S_t, \tilde{v}_t)^\top$, $W_t := (W^1_t, W^2_t)^\top$, $a(\theta, Y_t(\theta)) := \left( \frac{(r-\delta)S_t}{\kappa \tilde{v}_t^+} \right)$ and $b(\theta, Y_t(\theta)) := \left( \frac{\sqrt{\nu_0} \sqrt{\tilde{v}_t}}{\sqrt{\nu_0} \sqrt{\rho}} \frac{\nu_0}{\sqrt{\nu_0} \sqrt{\rho}} \sqrt{\nu_0} \sqrt{\tilde{v}_t} (1-\rho^2) \right)$ we can write the system (27) as

$$dY_t(\theta) = a(\theta, Y_t(\theta)) dt + b(\theta, Y_t(\theta)) dW_t$$

(28)

By the fundamental theorem of asset pricing, arbitrage-free prices of European call options with strikes $K_i$ and maturities $T_i$ are given by the discounted expected payoff under an equivalent martingale measure $Q$, i.e.

$$C^i(\theta) = e^{-rT_i} \mathbb{E}_Q [\max(S_{T_i}(\theta) - K_i, 0)]$$

(29)

We calibrate the model (28) to market prices $C^i_{\text{obs}}$ by minimizing the error functional

$$\text{Err}(\theta) = \sum_{i=1}^I \left( C^i(\theta) - C^i_{\text{obs}} \right)^2$$

(30)

over $\theta \in \Theta$ subject to (28) with $0 \leq t \leq T$, $T = \max_{i=1,...,I} T_i$.

In the Heston model the expectations in (29) can be computed by a semi-closed form solution, however in more general models we can only give estimators $\hat{C}^i(\theta)$ for $C^i(\theta)$, for example by using (quasi-)Monte Carlo simulation. We will describe this approach for the Heston model. As in the calibration of the one-factor Lévy gamma model in Section 5, the analytic formula for the expectations allows us to compare the quality of the optimal parameters obtained by calibrating with fixed Monte Carlo resp. quasi-Monte Carlo point sets in our numerical experiments.

In order to calibrate the Heston model to market data we discretize the system (28) in time with equidistant step size $\Delta t$ which leads to

$$y^{n+1}_m(\theta) = y^n_m(\theta) + a(\theta, y^n_m(\theta)) \Delta t + b(\theta, y^n_m(\theta)) \Delta W^n_m$$

$$y^n_0 = Y_0$$

$$m = 0, \ldots, M-1, \quad M = \frac{T}{\Delta t}, \quad n = 1, \ldots, N$$

Here $y^n_m(\theta)$ denotes the approximation of the $n$th path of $Y_{t_m}(\theta) = Y_{m\Delta t}(\theta)$. The two components of the vector $\Delta W^n_m$ are normally distributed random variables with mean $\mu = 0$ and variance $\Delta t$, so for the simulation of the $n$th path we need $2M$ pseudo random numbers $\mathbf{x}_n = (x_1, \ldots, x_{2M})^\top$ with $x_k \sim \mathcal{N}(0, \Delta t)$. 


With a fixed point set \((x_1, \ldots, x_N)\) our calibration problem leads to the optimization problem

\[
\min_{\theta \in \Theta} \hat{\text{Err}}(\theta) = \sum_{i=1}^{I} (\hat{C}^i(\theta) - C_{\text{obs}}^i)^2
\]

where

\[
C^i(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left( \pi_{\varepsilon} \left( s_{M_i}^n(\theta) - K_i \right) \right), \quad M_i = \frac{T_i}{\Delta t}
\]

subject to

\[
y_{m+1}^n(\theta) = y_m^n(\theta) + a_{\varepsilon}(\theta, y_m^n(\theta)) \Delta t + b_{\varepsilon}(\theta, y_m^n(\theta)) \Delta W_m
\]

Here \(s_{M_i}^n(\theta)\) is the first component of \(y_m^n(\theta)\) and the non-differentiable maximum function \(\max(x, 0)\) is approximated by the smooth function

\[
\pi_{\varepsilon}(x) := \begin{cases} 
0, & x < -\varepsilon \\
-\frac{1}{16\varepsilon}x^4 + \frac{3}{8\varepsilon}x^2 + \frac{1}{2}x + \frac{3\varepsilon}{16}, & -\varepsilon \leq x \leq \varepsilon \\
x, & x > \varepsilon
\end{cases}
\]

as proposed in [22]. In the same way \(a\) and \(b\) are smoothed by (32), which leads to continuously differentiable functions \(a_{\varepsilon}\) and \(b_{\varepsilon}\).

The accuracy of this simulation-based calibration certainly depends on \(N, \Delta t\) and \(\varepsilon\). Under which conditions the discretized optimization problem (31) converges to the solution obtained by minimizing the original problem (30) is studied in [22]. The smoothing out of the nondifferentiabilities allows us to make use of gradient-based optimization algorithms where we compute the gradient and the an approximation of the Hessian of the error functional \(\hat{\text{Err}}(\theta)\) by an adjoint method. Glasserman and Giles [17] have demonstrated that such an approach can significantly reduce computational time.

6.2 Numerical results

We calibrated the Heston model to a set of 100 European call options on the S&P 500 index taken from Andersen and Brotherton-Ratcliffe [4] with fixed risk-free interest rate \(r = 0.06\), dividend yield \(\delta = 0.0262\) and initial stock price normalized to \(S_0 = 1\). The market data is given in Table 2 in the form of implied volatilities.

As in the calibration of the credit risk model we compared the performance of four different fixed Monte Carlo point sets with a point set constructed by the Sobol sequence for \(N = 2^k, k = 1, 2, 3, 4\) points with time discretization \(\Delta t = 0.05\) and \(\varepsilon = 0.009\). As a reference value for quality of the estimated parameters we used the optimal parameter set from the closed-form solution

\[
\theta_{cfs}^* = (0.0146, 1.4377, 0.0245, 0.2655, -0.7305)^T.
\]

We minimized the optimization problem (31) by a sequential quadratic programming algorithm, where the gradient and the Hessian are computed by an adjoint method. Analogous to [22] we used the following bounds for the parameters

\[
0.01 < \theta_i < 100, \quad 0 < \Delta t < 0.1, \quad 0 < \varepsilon < 0.1
\]
<table>
<thead>
<tr>
<th>KST</th>
<th>0.175</th>
<th>0.425</th>
<th>0.695</th>
<th>0.94</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>0.190</td>
<td>0.177</td>
<td>0.172</td>
<td>0.171</td>
<td>0.171</td>
<td>0.169</td>
<td>0.169</td>
<td>0.168</td>
<td>0.168</td>
<td>0.168</td>
</tr>
<tr>
<td>0.90</td>
<td>0.168</td>
<td>0.155</td>
<td>0.157</td>
<td>0.159</td>
<td>0.159</td>
<td>0.160</td>
<td>0.161</td>
<td>0.162</td>
<td>0.164</td>
<td></td>
</tr>
<tr>
<td>0.95</td>
<td>0.133</td>
<td>0.138</td>
<td>0.144</td>
<td>0.149</td>
<td>0.150</td>
<td>0.151</td>
<td>0.153</td>
<td>0.155</td>
<td>0.157</td>
<td>0.159</td>
</tr>
<tr>
<td>1.00</td>
<td>0.113</td>
<td>0.125</td>
<td>0.133</td>
<td>0.137</td>
<td>0.138</td>
<td>0.142</td>
<td>0.145</td>
<td>0.149</td>
<td>0.152</td>
<td>0.154</td>
</tr>
<tr>
<td>1.05</td>
<td>0.102</td>
<td>0.109</td>
<td>0.118</td>
<td>0.127</td>
<td>0.128</td>
<td>0.133</td>
<td>0.137</td>
<td>0.143</td>
<td>0.148</td>
<td>0.151</td>
</tr>
<tr>
<td>1.10</td>
<td>0.097</td>
<td>0.103</td>
<td>0.104</td>
<td>0.113</td>
<td>0.115</td>
<td>0.124</td>
<td>0.130</td>
<td>0.137</td>
<td>0.143</td>
<td>0.148</td>
</tr>
<tr>
<td>1.15</td>
<td>0.120</td>
<td>0.100</td>
<td>0.100</td>
<td>0.106</td>
<td>0.107</td>
<td>0.119</td>
<td>0.126</td>
<td>0.133</td>
<td>0.139</td>
<td>0.144</td>
</tr>
<tr>
<td>1.20</td>
<td>0.142</td>
<td>0.114</td>
<td>0.101</td>
<td>0.103</td>
<td>0.103</td>
<td>0.113</td>
<td>0.119</td>
<td>0.128</td>
<td>0.135</td>
<td>0.140</td>
</tr>
<tr>
<td>1.30</td>
<td>0.169</td>
<td>0.130</td>
<td>0.108</td>
<td>0.100</td>
<td>0.099</td>
<td>0.107</td>
<td>0.115</td>
<td>0.124</td>
<td>0.130</td>
<td>0.136</td>
</tr>
<tr>
<td>1.40</td>
<td>0.200</td>
<td>0.150</td>
<td>0.124</td>
<td>0.110</td>
<td>0.108</td>
<td>0.102</td>
<td>0.111</td>
<td>0.123</td>
<td>0.128</td>
<td>0.132</td>
</tr>
</tbody>
</table>

Table 2: Market data: Implied volatilities of 100 European Call options on the S&P 500 index

\[
0.0001 \leq \mu_0 \leq 2, \quad 0.05 \leq \kappa \leq 2, \\
0.0001 \leq \theta \leq 2, \quad 0.0001 \leq \sigma \leq 4, \\
-0.985 \leq \rho \leq 0.985
\]

Furthermore the parameters \( \theta \) have to satisfy the so-called Feller condition \( \kappa \theta - \sigma^2 / 2 \geq 0 \), which guarantees that the variance process attains zero with probability zero. That constraint was directly included into our discretized problem by a parameter transformation, using \( \lambda = \sigma / \sqrt{2\kappa \theta} \) instead of \( \sigma \) with the bounds \( 0.0001 \leq \lambda \leq 0.9999 \).

In the quasi-Monte Carlo setting we have combined the Sobol sequence with the Brownian Bridge Construction. This approach was proposed by Moskowitz and Caflisch [31] and was found to be effective in finance applications by several authors (among others [1], [2] or [11]). In the present setup we could also observe an apparent improvement in the error of the estimated parameter with this approach.

Results of the calibration are given in Figures 11 to 15, where the errors in the parameters \( \theta \) are plotted.
Figure 11: Common (i.e. base-10) logarithm of the relative error in $\nu_0$ in dependence of the number of simulations $N$ for the calibration to European call options on the S&P 500 index.

Figure 12: Common logarithm of the relative error in $\kappa$ in dependence of the number of simulations $N$ for the calibration to European call options on the S&P 500 index.
Figure 13: Common logarithm of the relative error in $\theta$ in dependence of the number of simulations $N$ for the calibration to European call options on the S&P 500 index.

Figure 14: Common logarithm of the relative error in $\sigma$ in dependence of the number of simulations $N$ for the calibration to European call options on the S&P 500 index.
Figure 15: Common logarithm of the relative error in $\rho$ in dependence of the number of simulations $N$ for the calibration to European call options on the S&P 500 index.

From the plots we see that, on average, the calibration with Sobol point-sets is superior to that with fixed Monte-Carlo point-sets especially for small $N$. Altogether we can obtain similar results as in the calibration of the one-factor Lévy gamma model. Some of the point sets generated by pseudo random numbers give a smaller deviation in some parameters, however in practice one has no guarantee that the particular point set used is one of the good ones.

In order to eliminate the influence of the discretization bias we also calibrated the Heston model to hypothetical market data by fixing $\theta_{\text{hyp}} = (0.02, 1.50, 0.03, 0.28, -0.70)^\top$ and computing $C_{\text{obs}}^i$ by a Monte Carlo simulation in order to eliminate the errors of the model. However the results we obtained for that calibration did not differ much from those above.

7 Conclusions

We have discussed the advantages of using fixed random point sets when calibrating simulation-based models to market data. We have argued that for both theoretical and practical reasons low-discrepancy point sets should be used instead of randomly chosen ones. An error estimate for the calibrated parameters has been given under very strong assumptions, and numerical examples have been provided which confirm our prior arguments under relaxed assumptions.

The examples show that even for moderately non-smooth integrands the low-discrepancy sequence gives better results than a randomly chosen one.

Future work will encompass the search for error estimates under weaker assumptions than those in Section 4. Further one may look for low-discrepancy sequences with properties tailored to the pricing/calibration problem in question.
Acknowledgements The authors wish to thank Gerhard Larcher for valuable comments.

References


