



BUSINESS PERSPECTIVES



Publisher:

LLC "CPC "Business Perspectives"
Hryhorii Skovoroda lane, 10,
Sumy, 40022, Ukraine
www.businessperspectives.org



V. HETMAN KNEU



Founder:

State Higher Educational
Establishment "Kyiv National
Economic University named after
Vadym Hetman",
Prospect Peremogy, 54/1,
Kyiv, 03057, Ukraine
<https://kneu.edu.ua/>

Received on: 30th of
August, 2018

Accepted on: 17th of
October, 2018

© Maksym Bondarenko,
Victor Bondarenko, 2018

Maksym Bondarenko, Master's
Degree in International Econom-
ics, Master's Degree in Quantitative
Finance and Risk Management,
Consultant Developer C++, CapFi
Groupe, France.

Victor Bondarenko, Ph.D.
(Technical Sciences), Docent,
Associate Professor of Department
of Design of Electronic
Computational Equipment,
National Technical University
of Ukraine «Igor Sikorsky Kyiv
Polytechnic Institute», Ukraine.



This is an Open Access article,
distributed under the terms of the
[Creative Commons Attribution 4.0
International license](https://creativecommons.org/licenses/by/4.0/), which permits
unrestricted re-use, distribution,
and reproduction in any medium,
provided the original work is
properly cited.

Maksym Bondarenko (France), Victor Bondarenko (Ukraine)

CALIBRATION OF DUPIRE LOCAL VOLATILITY MODEL USING GENETIC ALGORITHM OF OPTIMIZATION

Abstract

The problem of calibration of local volatility model of Dupire has been formalized. It uses genetic algorithm as alternative to regularization approach with further application of gradient descent algorithm.

Components that solve Dupire's partial differential equation that represents dynamics of underlying asset's price within Dupire model have been built. This price depends in particular on values of volatility parameters. Local volatility is parametrized in two dimensions (by Dupire model): time to maturity of the option and strike price (execution price). On maturity axis linear interpolation is used while on strike axis we use B-splines. Genetic operators of mutation and selection are then applied to parameters of B-splines.

Resulting parameters allow us to obtain the values of local volatility both in knot points and intermediate points using interpolation techniques. Then we solve Dupire equation and calculate model values of option prices.

To calculate cost function, we simulate market values of option prices using classic Black-Scholes model.

An experimental research to compare simulated market volatility and volatility obtained by means of calibration of Dupire model has been conducted. The goal is to estimate the precision of the approach and its usability in practice.

To estimate the precision of obtained results we use a measure based on average deviation of modeled local volatility from values used to simulate market prices of the options.

The research has shown that the approach to calibration using genetic algorithm of optimization requires some additional manipulations to achieve convergence. In particular, it requires non-uniform discretization of the space of model parameters as well as usage of De Boor interpolation. Value 0.07 turns out to be the most efficient mutation parameter. Using this parameter leads to quicker convergence. It has been proved that the algorithm allows precise calibration of local volatility surface from option prices.

Keywords

genetic algorithm, stochastic optimization, local volatility, implied volatility, calibration, partial differential equations, Black Scholes model, Dupire model

JEL Classification

C15, C61, G12

М.В. Бондаренко (Франція), В.М. Бондаренко (Україна)

КАЛІБРУВАННЯ МОДЕЛІ ЛОКАЛЬНОЇ ВОЛАТИЛЬНОСТІ ДЮПІРА ІЗ ЗАСТОСУВАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ ОПТИМІЗАЦІЇ

Анотація

Формалізовано задачу калібрування моделі локальної волатильності Дюпіра із застосуванням генетичного алгоритму оптимізації, як альтернативу підходу «регуляризації» із подальшим використанням алгоритму градієнтного спуску.

Побудовано компоненти для розв'язання диференційного рівняння Дюпіра, яке відображає динаміку ціни на базовий актив в рамках моделі Дюпіра. Така ціна, окрім іншого, залежить від значень параметрів локальної волатильності, яку параметризовано за двома вимірами (за моделлю Дюпіра): часу до експірації опціону та ціною страйк (ціною виконання). За всією часу використано лінійну інтерполяцію, а за всією страйк – B-сплайни. До параметрів B-сплайнів

застосовано генетичні оператори селекції та мутації.

Результуючі параметри дозволяють отримати значення локальної волатильності у вузлових точках, а також в проміжних точках шляхом інтерполяції. Після цього шляхом розв'язку рівняння Дюпіра отримуються модельні значення цін на опціони.

Для розрахунку цільової функції промодельовано ринкові значення цін на опціони з використанням класичного варіанту моделі Блека-Шоулза.

Проведено експериментальне дослідження з порівняння модельованої ринкової волатильності та волатильності, отриманої шляхом калібрування моделі Дюпіра, для оцінки ефективності підходу і аналізу можливості його використання на практиці.

Для оцінки точності отриманих результатів використано міру, що базується на середньому відхиленні модельованої локальної волатильності, отриманої шляхом калібрування моделі, від реальних значень ринкових цін на опціони.

Дослідження показало, що підхід до калібрування з використанням генетичного алгоритму оптимізації вимагає застосування додаткових маніпуляцій для досягнення збіжності алгоритму, зокрема використання нерівномірної дискретизації простору параметрів моделі, а також алгоритму інтерполяції Де Бура. Виявлено найбільш ефективне значення параметру мутації для даної задачі, яке дорівнює 0.07. За цього значення збіжність алгоритму досягається найшвидше. Доведено, що алгоритм здатен досить точно калібрувати поверхню локальної волатильності з ринкових цін на опціони.

Ключові слова

генетичний алгоритм, стохастична оптимізація, локальна волатильність, імпліцитна волатильність, калібрування, рівняння в частинних похідних, модель Блека-Шоулза, модель Дюпіра

Класифікація JEL

C15, C61, G12

INTRODUCTION

Since the famous Black-Scholes model was introduced in 1973, option markets have evolved to become autonomous, organized markets with a fairly high degree of liquidity, especially for index options and foreign exchange options. In such markets, the market prices of a series of liquid options, which are often call or put options, are readily observed. These market prices are then used as a benchmark to “mark to market” or calibrate an option pricing model, which can then be used to compute prices of more complex (“exotic”) options or compute hedge ratios (Ben-Hamida, & Cont, 2013).

Determining of underlying asset's volatility from the set of observed market prices is called “model calibration” problem. It is the inverse problem to the option pricing problem.

Since the financial crash of 1987 the practitioners have noticed that the constant parameter of volatility calibrated as implied volatility from standard Black-Scholes model is not relevant anymore. In reality the volatility varied among different strikes and maturity dates for benchmark options. It has led to development of generalized models, including local and stochastic volatility models.

In the case of parametric approach, the volatility parameter is defined as finite dimensional vector (for local volatility, stochastic volatility or CEV model). Calibration problem lays in finding such value of parameters that the set of prices obtained from the model matches the set of observed market prices.

However, in practice the market information is not sufficient to completely identify a pricing model. If the model is sufficiently rich, several sets of model parameters can be compatible with market prices, leading to model ill-posedness and uncertainty (Ben-Hamida, & Cont, 2013).

In practice calibration problem is introduced as optimization problem: the goal is to match the market prices as close as possible. The common measure of discrepancy between model prices and observed market prices is a sum of squared differences between them. The weighted cost function is defined as following:

$$G(\theta) = \sum_{i=1}^I \left| Call^{model}(t, S_t T_i, K_i) - Call_t^{market}(T_i, K_i) \right|^2 w(i), \quad (1)$$

where S – spot price of the underlying asset; K – strike price of the option; T – maturity of the option; w – weight of the given option in cost function; t – time at which we price the option (normally today's date).

To minimize this cost function, the classical gradient methods could be used. However, the cost function is nei-

ther convex nor does it have any particular structure enabling the use of gradient-based minimization methods to locate the minima (Ben-Hamida, & Cont, 2013). Moreover, the function is not given explicitly and solutions are obtained by solving a partial differential equation or executing a Monte-Carlo simulation. Finally, we are not even sure that the gradient-based algorithm will not fall in a local minima instead of a global one.

There are several approaches to regularize the ill-posed problem as well as introduce some convexity penalization criteria to ensure the ability of gradient-based algorithm to find a well-matching set of parameters. The motivation of this work is to implement alternative approach introduced by Ben-Hamida and Cont that does not suppress the ill-posed character of the problem but rather reflects it. The genetic optimization algorithm is used to overcome the problems of non-convexity and multiplicity of local minima.

1. LITERATURE REVIEW

Current research is inspired by the results of Ben-Hamida and Cont paper (2013). The authors introduce a probabilistic approach for calibration of local volatility surface from the set of observed market prices. Calibration involves usage of genetic algorithm: its steps are applied to local volatility surface presented as cubic B-Spline on strike axis and linear interpolation on maturity axis. The parameters of B-Spline are exposed to cycles of genetic transformations that lead to obtaining the fittest surface.

Ben-Hamida and Cont in their article claim that Dupire formula for obtaining local volatility from the market prices is unstable: Dupire (1994) presents a formula for reconstructing local volatility functions from a continuum of call option prices; however, this formula involves taking derivatives from discrete data and is numerically unstable. A discretized version of the Dupire formula is the implied tree method of Derman et al (1996), which is prone to similar instabilities leading to “negative probabilities” (Ben-Hamida, & Cont, 2013). That is why Ben-Hamida and Cont describe an alternative way to calibrate the local volatility directly.

Cerf (1997) and Del Moral (2001) introduce the main steps of particle genetic algorithm and prove its convergence. Cerf proposes to decrease the temperature parameter while moving through algorithm’s iterations. He suggests that this (progressive increase of selection pressure) will lead the population to concentrate around the global minima. Alternative implementation of natural (binary) genetic algorithm is described by Dieterle (2003), Razali and Geraghty (2011). Both authors focus on the importance of selection procedure in binary approach. For this research we have nevertheless chosen the particle algorithm as it requires less computational cost in general case.

Bonnans, Cognet and Volle (2002) use the Newton-Raphson algorithm for calibration of local volatility and regularize the problem by using the same regularization term as Lagnado and Osher (1997). As an alternative they introduce another way to regularize the problem – multiscale approach. Howison (1999) claims that the regularization method provided by Lagnado and Osher (1997) has some drawbacks among which is computational cost. Ben-Hamida and Cont, however, reject usage of regularization as it is not necessary for direct calibration of local volatility. In this research we do not add any regularization terms to the cost function.

Bonnans, Cognet and Volle also introduce a usage of implicit schema to solve the PDE of Dupire that permits us to obtain model prices. In (Bonnans, Cognet & Volle, 2002) the authors prove that usage of implicit schema with uniform discretization schema will give incoherent results for the near-the-strike prices and non-uniform discretization will permit us to obtain more precise prices near-the-strike. We will use the same non-uniform schema in our research. Howison (1999) as well as Ben-Hamida and Cont uses weighted minimization (weight of near-the-strike prices in computation of cost function is bigger than the weight of less tradable options). The results of this research are obtained by using weighted approach as well.

Li and Coleman (1999) introduce the cubic spline approximation of local volatility function. In our research however we still use its representation in B-splines proposed by Ben-Hamida and Cont (2013) as it is well adjusted to the regularization approach that they use. In future work we suggest usage of De Boor’s algorithm of spline interpolation as it reflects better the optimization problem and possibly can improve the convergence.

2. RESEARCH OBJECTIVES

The objective of current research is to introduce the state of the art in local volatility calibration problem and implement the algorithm to solve this problem. We aim to obtain coherent results of calibration avoiding at the same time high dimensionality of the problem and reducing the computational costs by applying interpolation techniques. We are also willing to enhance the existing approach by improving algorithm's components. Introducing and calculating a measure of quality (Section 3.3) of the calibration is a novelty of current work.

Firstly, we aim to produce a general view on local volatility calibration and introduce a formal representation of the problem including the cost function. We will also do a brief overview of particle genetic algorithm.

Then we introduce Dupire framework and challenges that raise up within it. The goal is to provide methodological guide to implement next brick of the algorithm: Dupire differential equation solver that is used to calculate the cost function on every iteration of genetic algorithm.

Finally, we introduce the requirements for the input of optimization problem – local volatility surface. We also provide a guideline to create a smooth representation of the local volatility surface using B-Spline interpolation with uniform knots.

In the last section we provide numerical results obtained by running the genetic algorithm to calibrate local volatility surface. We estimate a quality of the approach using an average-based measure.

The global goal of this research is to prepare the basis for the following improvements in the analysis of predictive power of volatility. Obviously the set of obtained model parameters can be used to study the predictive ability of local volatility by regressing local volatility parameter as dependent variable on a time variable. This will be the next step to introduce a neural network constructed from a set of regression models correspondent to different volatility models (local, stochastic, implied volatility etc.). Such a network can potentially improve banks' forecasts for realized market volatility and thus improve the pricing accuracy. It can be also used for overnight risk reports and pricing valuation adjustments that require pricing of options at future times.

3. THEORETICAL BASICS

3.1. Calibration problem setting

The optimization problem, where the input is a parameterized local volatility surface is:

$$\inf_{\theta \in E} G(\theta), G(\theta) = \sum_{i=1}^{discT} \sum_{j=1}^{discK} |C^\theta(t, S_i, T_i, K_j) - C_i^*(T_i, K_j)|^2 w_{i,j}, \quad (2)$$

where C^* – a market observed price of a Call; C^θ – a theoretical price obtained for a set of market parameters and a local volatility surface corresponding to a set of parameters θ ; $w_{i,j}$ – the weights determined by the fact how for the option is in- or out-of-money; $discT$, $discK$ – number of values (discretization) of time and strike axis respectively.

As the function $G(\cdot)$ is neither convex nor continuous, the gradient based approaches will not work in the regularization of the problem.

The function $G(\cdot)$ is not given explicitly as well. The solution for C^θ is obtained by solving the Dupire PDE (Section 4.2) for the corresponding set θ . To regularize the input (smooth volatility surface), we use B-Spline representation in Section 5.1.

Until the stopping criteria is achieved we improve the set of values C^θ by adjusting the parameter set θ using the genetic algorithm (Section 3.2) as one that does not requires any gradient calculations and reduces the dimension of the problem. Adjusting of parameter θ allows us to obtain a new (improved) smooth representation of a function $\sigma(K, T)$ that is now a local volatility function.

3.2. Particle Genetic Algorithm

Genetic algorithms are stochastic search methods based on natural evolution processes. They are defined as a system of particles (or individuals) evolving randomly and undergoing adaptation in a time non necessarily homogeneous environment represented by a collection of fitness functions (Del Moral, & Miclo, 2001).

The simplest genetic algorithm is a two-stages and time in-homogeneous Markov chain given for each $n \geq 0$ by setting:

$$\xi_n = (\xi_n^1 \dots \xi_n^N) \xrightarrow{\text{Selection}} \hat{\xi}_n = (\hat{\xi}_n^1 \dots \hat{\xi}_n^N) \xrightarrow{\text{Mutation}} \xi_{n+1}. \quad (3)$$

Here we have:

- integer N represents the size of population;
- the initial system $\xi_0 = (\xi_0^1 \dots \xi_0^N)$ consists of N independent random particles;
- gibbs measure is introduced as $G(\xi_n^i) = \exp(-V(\xi_n^i)/T)$, where $V(\xi_n^i)$ is a cost function and T is a temperature parameter;
- in the selection transition the particles $\hat{\xi}_n = (\hat{\xi}_n^1 \dots \hat{\xi}_n^N)$ are chosen randomly and independently of previous configuration according to a given measure (Gibbs) $G(\xi_n)$ with probability of selection of the i -th individual equals $P(\hat{\xi}_n^i = \xi_n^i | \xi_n) = G(\xi_n^i)$ and probability of its replacement by another individual equals

$$P(\hat{\xi}_n^i = \xi_n^j | \xi_n) = \frac{G(\xi_n^j)}{\sum_{i=1}^N G(\xi_n^i)};$$

- mutation Kernel can vary however we use the simplest representation $\xi_{n+1}^i = \xi_n^i + t(2 * rand() - 1)$;
- t is a step (small value) and T is decreasing with increasing number of iterations;
- we run algorithm until at least one member of a population ξ_n will satisfy stopping criteria (cost function will be minimized) i.e. sum of square differences will be sufficiently small.
- Genetic algorithm is used in practice for several reasons:
- it does not requires differentiability of the cost function;
- alternatively, to Newton-Raphson algorithm falls to the global minima when the function is not strictly convex;
- converges relatively fast.

The local volatility calibration problem falls into these categories.

3.3. Solution quality criterion

We consider the calibration successful if the average value of volatilities calibrated for at-the-money options (i.e. such that the strike equals to the spot price of the underlying) at every maturity lays in 30% threshold with respect to the value $\sigma = 0.2$ (we use $\sigma = 0.2$ for simulating market price of the option at every point (i, j) and for the sake of simplicity it is taken constant). In the case of absolute convergence $\sigma(\cdot, at) = 0.2$, where at is an index of at-the-money option. The simple average:

$$deviation(at) = \frac{\sum_{i=1}^{discT} \sigma(i, at) - 0.2}{discT - 0.2}. \quad (4)$$

And the quality measure:

$$Q_1(\text{deviation}) = I(\text{deviation} < 0.3), \quad (5)$$

where $I(\cdot)$ – an indicator function returning 1 if the argument is true (deviation is small enough and quality is sufficient for given strike) or 0 if argument is false (quality is insufficient).

However previous argument can be too weak to represent overall quality of the algorithm as we are interested not specifically in local volatility calibrated from at-the-money options (one strike) but rather in volatilities from the set of close-to-the-money options. So we introduce new measure

$$Q = \frac{\sum_{k=at-\frac{discK}{10}}^{at+\frac{discK}{10}} Q_1(\text{deviation}(k))}{2 \frac{discK}{10}}. \quad (6)$$

Of course $\frac{discK}{10}$ is used only when $discK = 50$ in pair with $at = 25$ as it represents 10 prices close-to-the-money.

If we change discretization this value is to be changed (however in this paper, we take into account 10 values). Average-based measure Q is basically used with an old measure as an argument. We aim to obtain the output close to 1. $Q = 1$ corresponds to absolute quality, while $Q = 0$ corresponds to bad quality. We evaluate this measure in numerical results.

4. LOCAL VOLATILITY THEORETICAL BASICS

4.1. Implied and local volatilities

In the standard Black-Scholes model the volatility parameter is supposed to be constant with respect to maturity and strike. However, on the real market we observe that the volatility varies depending on both of the variables.

Generalized Black-Scholes framework permits us to solve the inverse problem for every maturity T and obtain the implied volatility values, however it still supposes that volatility is flat with respect to strike axis (i.e. for any maturity volatility is the same for any strike option), while we can observe the “smile” in the real market.

Dupire (1994) in his research introduces the local volatility as a function of time and stock price S_t which can be rewritten in terms of maturity and strike:

$$\sigma(T, K) = \sqrt{\frac{\frac{dC}{dT} + rK \frac{dC}{dK}}{2K^2 \frac{d^2C}{dK^2}}}. \quad (7)$$

Dupire formula for local volatility can be rewritten in terms of implied volatilities as well:

$$\sigma(T, K) = \sqrt{\frac{\sum^2 + 2 \sum T \left(\frac{d\sum}{dT} + rK \frac{d\sum}{dK} \right)}{1 + 2d_1 K \sqrt{T} \frac{d\sum}{dK} + K^2 T (d_1 d_2 \left(\frac{d\sum}{dK} \right)^2) + \sum \frac{d^2 \sum}{dK^2}}}. \quad (8)$$

or implementing $y = K/F(t, T)$:

$$\sigma(T, y) = \sqrt{\frac{\hat{\Sigma}^2 + 2\hat{\Sigma}T \frac{d\hat{\Sigma}}{dT}}{1 + 2d_1 y \frac{d\hat{\Sigma}}{dy} + y^2 T (d_1 d_2 (\frac{d\hat{\Sigma}}{dy})^2) + \hat{\Sigma} \frac{d^2 \hat{\Sigma}}{dy^2}}} \quad (9)$$

Being armed with Dupire formula one can proceed to calibration of local volatility surface having it. Among practitioners it is preferred to use the Dupire formula as presented in (7) – (9) and (12).

In this case the standard procedure to calculate the local volatility surface from the implied volatility one is next:

1. main requirements for the implied volatility surface are: C^1 continuity w.r.t. time axis, C^2 continuity w.r.t. strike axis. Thus to get a sufficiently smooth surface, we have to use linear interpolation on time axis and spline/2-nd or 3-rd order polynomial interpolation on strike axis;
2. calibrate the implied volatility independently for each pair (T, K) using a Black-Scholes formula and gradient-based optimization algorithm;
3. obtain the values of implied volatility at intermediate points (t, k) employing an interpolation method to the points obtained in the previous step;
4. use Dupire formula to obtain the local volatility from now differentiable (smooth) implied volatility surface.

However, Ben-Hamida and Cont (2013) claim the instability of Dupire's formula based methods: Dupire presents a formula for reconstructing local volatility functions from a continuum of call option prices; however, this formula involves taking derivatives from discrete data and is numerically unstable. A discretized version of the Dupire formula is the implied tree method of Derman et al (Ben-Hamida, & Cont, 2013), which is prone to similar instabilities leading to “negative probabilities”.

Alternatively, we will discover their approach that avoids calculation of implied volatility and calibrate the local volatility surface directly using the genetic algorithm. The advantages of direct calibration of local volatility surface (Ben-Hamida, & Cont, 2013):

- no interpolation of option prices or implied volatilities is required. Contrary to implied tree (Derman et al) methods or methods based on the Dupire formula, it does not require call–put prices for all strikes or maturities nor any ad hoc interpolation of observed prices. They can therefore be applied to index options, where the number of observations is large and also to equity options for which less amount of data is present;
- avoidance of computing the high-dimensional gradient of the objective function;
- do not require convexity of the objective function;
- provides a population of solutions: it means that this approach rather reflects the ill-posedness of the problem and its uncertainty than eliminates it.

So instead of using the standard approach i.e. calibrate and interpolate the implied volatilities, we will work on direct calibration of local volatility surface by solving the Dupire PDE for each pair (T, K) being given the input parameter $\sigma(T, K)$. The surface $\sigma(T, K)$ will nevertheless be introduced as a combination of B-splines and the genetic algorithm will be applied not directly to $\sigma(K, T)$ but to parameters of the spline representing it. The usage of it will become evident in Section 4.3, where we introduce the Dupire pricer.

4.2. Dupire theory

We consider the Black-Scholes model of asset price evolution:

$$dS_t = S_t(\mu dt + \sigma dW_t), \quad (10)$$

where the volatility of the underlying is supposed to be constant. As volatility is the only non-observable parameter in this model, we can determine it for a quoted option (T, K) . The volatility obtained in this way is called

Implied volatility.

However, we can see that the volatility is pretty much dependent on strike and maturity of an option and is not constant at all (phenomena of “volatility smile”). That is why we look for another representation of local volatility. Many articles propose next diffusion process:

$$dS_t = S_t(\mu dt + \sigma(S_t, t)dW_t). \quad (11)$$

Dupire proposed a PDE that governs the price of the whole grid of the options. Local volatility $\sigma(S, t)$ is a coefficient of this PDE.

Dupire equation is formulated using the equation of Fokker-Planck. We consider case with no dividends. The Dupire equation:

$$\frac{dV}{dT}(K, T) = -rK \frac{dV}{dK}(K, T) + \frac{1}{2} \sigma^2(K, T) K^2 \frac{d^2V}{dK^2}(K, T), \quad (12)$$

with a set of conditions:

$$\begin{cases} V(K, t_0) = \max(S_0 - K, 0) \\ \lim_{K \rightarrow 0} V(K, T) = S_0 \\ \lim_{K \rightarrow \infty} V(K, T) = 0 \end{cases} \quad (13)$$

In Section 4.3 we present the numerical solution of Dupire equation.

4.3. Implicit schema solution

In the Section 4.1 we have described two ways to obtain the local volatility: the standard approach that uses Dupire formula to pass from implied volatilities to local volatilities and the direct calibration of local volatilities (implies solving the Dupire equation for every pair (T, K) of interest). As the first approach has some numerical instabilities we will use the direct calibration.

To obtain a solution of Dupire PDE for an input $\sigma(K, T)$ we have to pass through several steps:

1. logarithmic change of variable (transform the PDE and the set of conditions);
2. introduce new space discretization – we will use non-uniform discretization;
3. introduce time discretization – we use theta-schema with $\theta = 0$ that is equivalent to implicit discretization schema;
4. write the PDE in terms of new operators and obtain the tridiagonal system of equations;
5. obtain the resulting matrix of option prices by solving the tridiagonal system with Thomas algorithm.

The logarithmic change of variable (Bonnans, Cognet, & Volle, 2002): we do change of variable $y = \ln(K)$ and $U(y, T) = V(K, T)$. New system to solve:

$$\begin{cases} \frac{dU}{dT} + \tilde{A}U(y, T) = 0 \\ U(y_{\min}, T) = S_0 \\ U(y_{\max}, T) = 0 \\ U(y, t_0) = \max(S_0 - e^y, 0) \end{cases} \quad (14)$$

Grid discretization, where:

$$\tilde{A}_i(T) = \alpha_{i,T} u_{i-1}(T) + \beta_{i,T} u_i(T) + \gamma_{i,T} u_{i+1}(T). \quad (15)$$

With Uniform discretization case (Bonnans, Cognet, & Volle, 2002):

$$\alpha_{i,T} = -\frac{\sigma^2(y_i, T)}{2h^2} - \frac{1}{2h} \left(r + \frac{\sigma^2(y_i, T)}{2} \right)$$

$$\beta_{i,T} = \frac{\sigma^2(y_i, T)}{h^2}. \quad (16)$$

$$\gamma_{i,T} = -\frac{\sigma^2(y_i, T)}{2h^2} + \frac{1}{2h} \left(r + \frac{\sigma^2(y_i, T)}{2} \right)$$

Non-Uniform discretization case (Bonnans, Cognet, & Volle, 2002):

$$\alpha_{i,T} = -\frac{\sigma^2(y_i, T)}{(h_i + h_{i-1})h_{i-1}} - \frac{1}{2h_{i-1}} \left(r + \frac{\sigma^2(y_i, T)}{2} \right)$$

$$\beta_{i,T} = \frac{\sigma^2(y_i, T)}{h_i + h_{i-1}} \left(\frac{1}{h_i} + \frac{1}{h_{i-1}} \right) + \frac{1}{2} \left(r + \frac{\sigma^2(y_i, T)}{2} \right) \left(\frac{1}{h_{i-1}} - \frac{1}{h_i} \right), \quad (17)$$

$$\gamma_{i,T} = -\frac{\sigma^2(y_i, T)}{(h_i + h_{i-1})h_i} + \frac{1}{2h_i} \left(r + \frac{\sigma^2(y_i, T)}{2} \right)$$

where the ingredients are:

we define the function

$$g(y) = \frac{1}{2} \ln((y_{\max} + y) / (y_{\max} - y)) + y_0 \quad (18)$$

and let

$$h = 2y_{\max} / N = \delta \ln(K)$$

$$y_i = y_{\min} + i * \delta \ln(K) \quad (\text{uniform case}) \quad \text{or} \quad (19)$$

$$y_i = g(-y_{\max} + ih) = g(\ln(K_i)) \quad (\text{non-uniform case})$$

we define $h_i = y_{i+1} - y_i$.

Time discretization (theta-schema)

$$\left\{ \begin{array}{l} \frac{u_{n+1} - u_n}{k} + \theta(\tilde{A}^n u^n) + (1 - \theta) \tilde{A}^{n+1} u^{n+1} = 0 \\ U_0 = \max(S_0 - e^y, 0) \end{array} \right. \quad (20)$$

System to solve (implicit schema when $\theta = 0$)

$$\left\{ \begin{array}{l} H^n u^{n+1} = b^n \\ b^n = (I - \theta k \bar{A}^n) u^n - S_0 k (\theta_{\alpha 1, Tn} + (1 - \theta) \alpha_{1, T_{n+1}} e_1) \\ H^n = I + (1 - \theta) k \bar{A}^{n+1} \end{array} \right. \quad (21)$$

where I is a unary matrix (all values are equal to 1); k is a δT discretization step on time axis; e_1 is a vector with first value equal to 1 and all others equal to 0 (to reserve the initial condition's role).

The tridiagonal system representation

$$\hat{A}_T = \begin{pmatrix} \beta_{1,T} & \gamma_{1,T} & 0 & 0 & \cdots & 0 \\ \alpha_{2,T} & \beta_{2,T} & \gamma_{2,T} & 0 & \cdots & 0 \\ 0 & \alpha_{3,T} & \beta_{3,T} & \gamma_{3,T} & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \cdots & 0 & 0 & \alpha_{n-1,T} & \beta_{n-1,T} & \gamma_{n-1,T} \\ \cdots & 0 & 0 & 0 & \alpha_{n,T} & \beta_{n,T} \end{pmatrix}. \quad (22)$$

And $H^n u^{n+1} = b^n$ is presented as:

$$\begin{pmatrix} \beta_{1,T^{k+1}} & \gamma_{1,T} & 0 & 0 & \cdots & 0 \\ \alpha_{2,T} & \beta_{2,T^{k+1}} & \gamma_{2,T} & 0 & \cdots & 0 \\ 0 & \alpha_{3,T} & \beta_{3,T^{k+1}} & \gamma_{3,T} & 0 & \cdots \\ \cdot & \cdot & \cdot & \ddots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ddots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdots & 0 & 0 & \alpha_{n-1,T} & \beta_{n-1,T^{k+1}} & \gamma_{n-1,T} \\ \cdots & 0 & 0 & 0 & \alpha_{n,T} & \beta_{n,T^{k+1}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix}. \quad (23)$$

Solve the tridiagonal system of equations (Generalized Thomas algorithm)

Apply Thomas algorithm of diagonalization to solve the system for each time step n .

Let us put into the system the values of $\sigma(K, T)$ that we have used to simulate market prices. Finally, we obtain the surface of Dupire prices of the call (Figure 1). The discrepancy between the prices obtained and the market prices is illustrated in the Figure 2.

Usage of non-uniform schema allows us to obtain better prices around the strike (5-10% difference with real prices) when solving PDE. On Figure 2 we can see that the prices farther from the strike (out-of-the-money) are less precise and even non-coherent.

The values of $\sigma(K_p, T)$ used are equivalent to values $\sigma(y_p, T)$. Therefore, we are interested to interpolate these values at each point i to put into Dupire PDE

5. REPRESENTATION OF A LOCAL VOLATILITY SURFACE

5.1. B-splines representation of local volatility surface

We reconstruct the volatility surface by means of B-splines to achieve two goals:

- smooth representation of volatility surface (C^2 continuity);

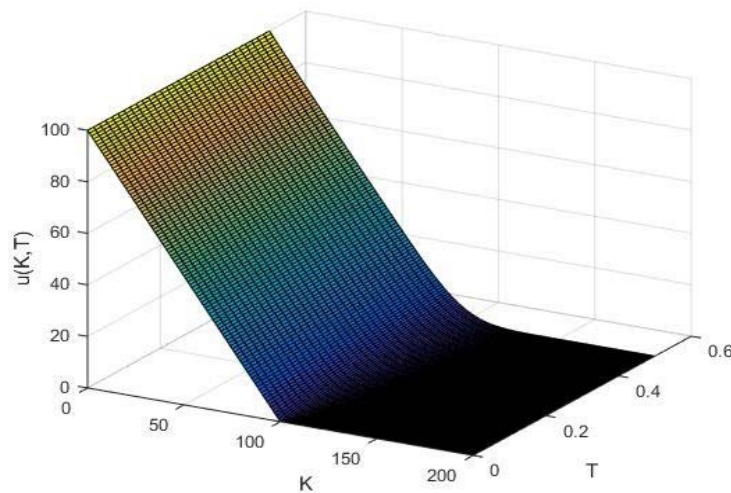


Figure 1. Dupire price surface

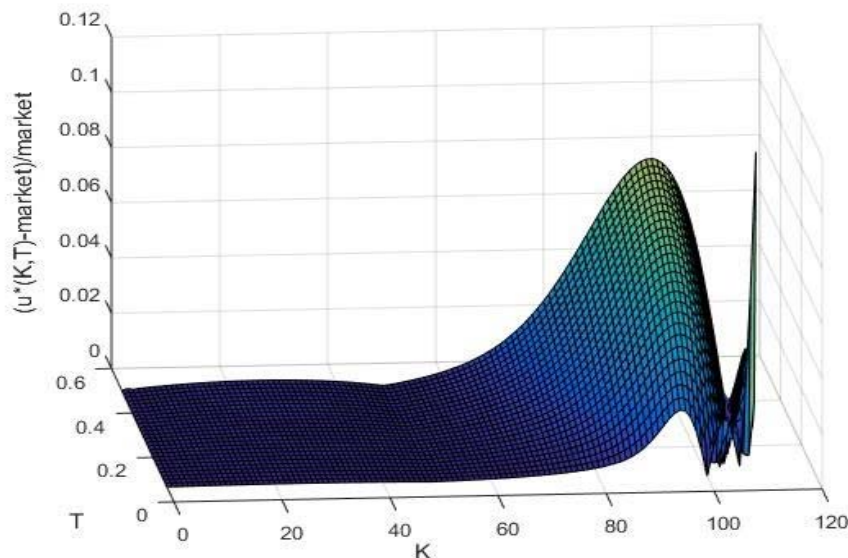


Figure 2. Discrepancy surface solver.

- decrease the scale of the problem (minimize the number of inputs to optimization algorithm).

Let us define the local volatility function as a smooth polynomial defined completely by basis functions associated with n points (“knots”) and $n - 4$ corresponding “control points” (values at certain knots). Number of control points is always equals: $m = n - p - 1$, where p is a degree of the interpolating polynomial. This means that 2 knots at each bound do not have a corresponding control point.

For our case we define K-grid representation:

- control points is a set of θ – parameters of volatility curve to be found;
- basis functions are defined for $p = 3$ case at each of the knot intervals. Number of knots n is smaller than the discretization of the grid (to achieve the smaller scale of the optimization problem);
- in Ben-Hamida and Cont (2013) representation knot intervals are equal

$$\sigma(T_i \cdot) = \sum_{m=1}^M \theta(i, m) \phi_m(\cdot). \quad (24)$$

We obtain the values on T-grid by means of linear interpolation:

$$\sigma(t, x) = \frac{T_{i+1} - t}{T_{i+1} - T_i} \sigma(T_i, x) + \frac{t - T_i}{T_{i+1} - T_i} \sigma(T_{i+1}, x). \quad (25)$$

To obtain the spline of 3-rd order we can use the basis functions introduced in Ben-Hamida and Cont (2013) or apply the generic De Boor’s algorithm to find Basis functions on uniform knots, which are presented in the Figure 3.

Let us reconstruct a pseudo-volatility function for next parameters:

- $X = [0 : 1 : 10]$ is a knot vector containing 11 uniformly distanced values;
- $x = rand(1,11)$ random control points corresponding to the knots;
- $p = 3$ degree of polynomial.

We want to evaluate the B-Spline in the points $k = 1 : 0.001 : 10$ and present it in Figure 4.

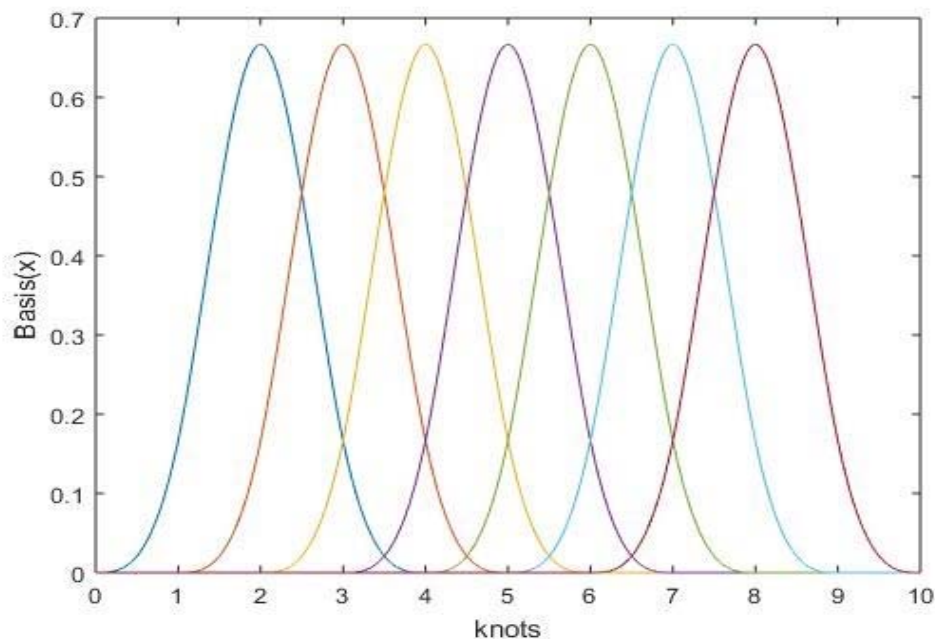


Figure 3. Basis functions on uniform knots

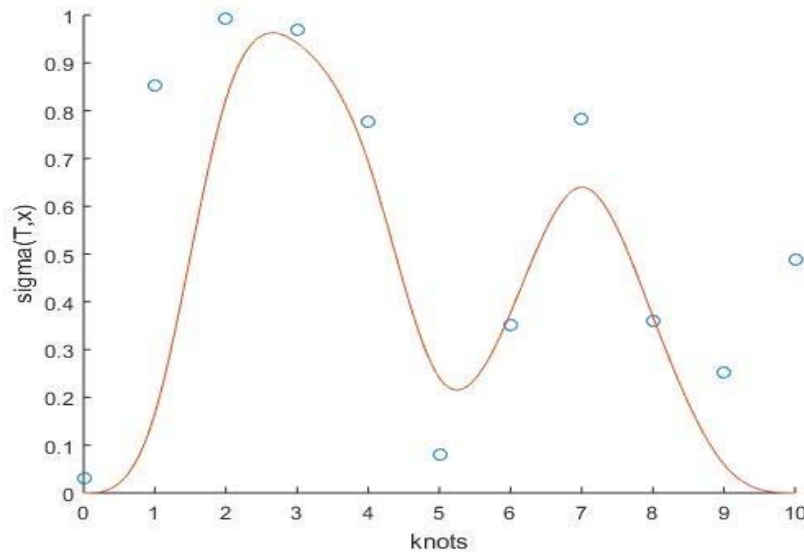


Figure 4. Spline evaluated at intermediate points

5.2. Ill-posedness and smoothness requirements (optional improvement)

The main difficulty with solving the inverse problem (calibration problem) arises when we are looking for an input to the Dupire pricer introduced in the Section 3.2. Basically we could start from any random points $\sigma(K,T)$ corresponding to the points known from market data. However, this naive approach has several drawbacks:

- the dimension of the problem will be equal to $N*M$, where N, M are sizes of discretization grid w.r.t. time and strike respectively. It will require an enormous computational cost (too much inputs into genetic algorithm);
- the ill-posedness of the problem causes the existence of infinite number of solutions and the naive representation will lead us to instability of the solution obtained.

Ill-posedness of the problem can be defined as non-continuity of the dependence of local volatility function on market data. It means that small perturbations in the price data will result in large changes in minimizing function and any solution obtained will be unstable (especially for the volatility calibrated for short maturities). Ill-posedness implies the non-uniqueness of solution as well.

To put some restriction on the solutions accepted Ben-Hamida and Cont introduce the smoothness requirement. Authors (Ben-Hamida, & Cont, 2013) propose to use smoothness semi-norm to quantify the smoothness of the local volatility surface:

$$\|\sigma\|^2 = \sum_i^T \int_{-A}^A dx \int_0^T \left| \frac{d\sigma}{dT} \right|^2 + \left| \frac{d^2\sigma}{dx^2} \right|^2 dx \quad (26)$$

or in matrix form:

$$\|\sigma\|^2 = \theta A \theta^T, \quad (27)$$

where θ is a matrix of control points in B-Spline representation.

Despite we will not use the smoothness requirement in our solution we will introduce the approach used by Ben-Hamida and Cont to construct the multivariate Gaussian random vector from the set of standard Gaussian random variables. It can be potentially used to create a set of Gaussian vectors applied in affine transformation of random mutation to improve the convergence and assure some acceptable only solutions.

Density of multivariate Gaussian by definition:

$$f(x) = \frac{1}{|2\pi A|} e^{-\frac{1}{2}(x-\mu)^T A^{-1}(x-\mu)}, \quad (28)$$

where $\det(A) \neq 0$ (A – is a covariance matrix and it has to be invertible).

Here $(x-\mu)^T A^{-1}(x-\mu)$ is a quadratic form equivalent to (15) in our case.

By the affine property any affine transformation of a Gaussian is a Gaussian i.e.

$$X \sim N(\mu, A) \Rightarrow BX + b \sim N(B\mu + b, BAB^T), \quad (29)$$

where X is a Gaussian vector.

Hence the random vector is constructed as following: let set of independent $X_1 \dots X_n \sim N(0,1) \Rightarrow X \sim N(0, I)$, where I is identity matrix. Then:

$$BX + \mu \sim N(\mu, A), \quad (30)$$

where $A = BB^T$.

The Gaussian B then can be used in mutation kernel that is equivalent to the affine transformation (18) with $\mu = 0$.

Matrix B has to be found by applying Cholesky factorization to positive semi-definite matrix A .

5.3. Algorithm description

In the Figure 5 we describe the workflow of our program. Firstly we generate the initial population of control points θ_m^i (as volatility surface is parametrized with control points of B-splines on strike axis and linear interpolation on time axis) for $i \in 0..I$ and $m \in 0..M$, where M – number of control points on strike axis and I – number of linear interpolation points on time axis.

Then for each member of population B-spline is built with help of such control points and $\sigma(t,x)_p$ is obtained for all $t \in 0..T$ and $x \in 0..K$, where T, K – discretization numbers by time and strike respectively, p – index of population member.

Then σ is inserted do Dupire equation solver and solution (prices) $C(t,x)_p$ is obtained for each t, x on discretization grid for each p .

Prices are compared to real market prices for each t, x and for each p and the sum of squared differences is considered to measure the fitness of each p -th candidate. If the desired fitness is not found, we pass to next step otherwise program terminates.

Candidate matrices of control points $\theta_m^i(p)$ then are subject to genetic transformations (selection and mutation) based on the fitness of corresponding solutions for each p . New population is formed and algorithm repeats.

Remark: The market data is simulated from Black-Scholes formula to achieve better flexibility in testing calibration for different values of parameter σ .

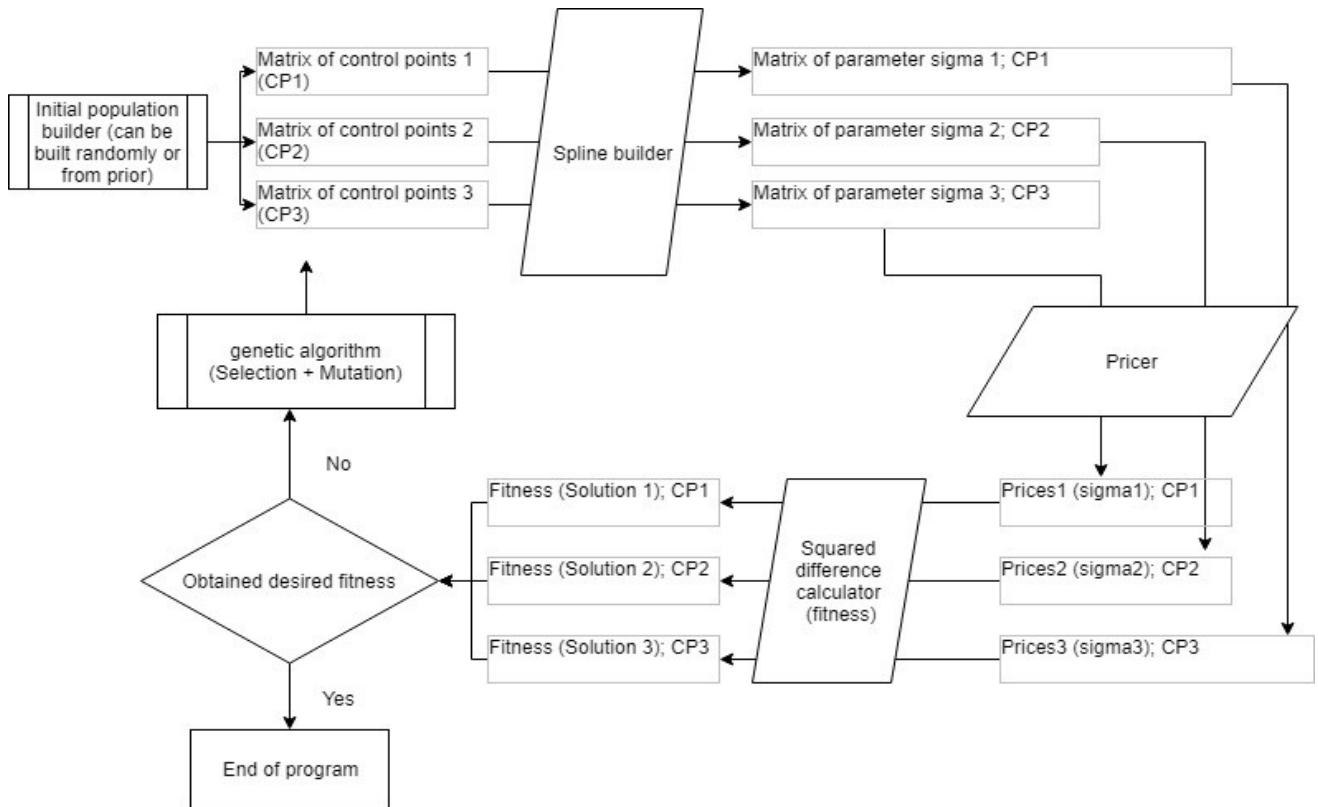


Figure 5. Program workflows

5.4. Algorithm setting

Model settings:

$$K \in [0.005; \quad 0.005 + \delta k \quad ..200], \quad (31)$$

$$\text{where } \delta k = \frac{(200 - 0.005)}{\text{disc}K}.$$

$$T \in [0; \quad 0 + \delta k \quad ..5], \quad (32)$$

where $\delta k = \frac{(5 - 0)}{\text{disc}T}$; discK = 50 (strike axis discretization); discT = 20 (time axis discretization); S = 100 (spot price); r = 0 (rate).

Genetic algorithm settings:

- $T_0 = 1000$ (initial temperature);
- $t = 0.03, 0.05, 0.07$ (mutation rate varying for experiments);
- $N_{mc} = 2^{14}$ (maximum number of simulations);
- $M = 64$ (temperature decreasing frequency);
- $p = 20$ (population size);
- stopping criteria: $G(\theta) < 15$.

Miscellaneous:

- market data $C_t^*(T_p, K_p)$ is simulated from: $\sigma = 0.2$ for all K, T using Black-Scholes model;
- quality criteria: we use measure Q (Section 3.3) to qualify obtained results.

5.5. Fitness function

The weighted sum of squared differences was chosen as cost (fitness) function to minimize. The sum is taken over all strike/maturity pairs indexed $i \in [1..DiscT]$, $j \in [1..DiscK]$ respectively and weighted on weight function:

$$w(i, j) = \left(\frac{1}{0.1 * \sqrt{2\pi}} \right) * \exp\left(-0.5\left(\frac{x}{0.1}\right)^2\right), \quad (33)$$

$$\text{where } x = \frac{(j - DiscK / 2)}{DiscK}.$$

If we refer to function (2) the weight $w_{i,j}$ is obtained as:

$$w_{i,j} = \frac{weight(i, j)}{\sum_{jj=1}^{DiscK} weight(i, jj)}. \quad (34)$$

Then we get to the following form of the cost function:

$$\inf_{\theta \in E} G(\theta), G(\theta) = \sum_{i=1}^I \sum_{j=1}^J \left| C^\theta(t, S_t, T_i, K_j) - C_t^*(T_i, K_j) \right|^2 \frac{weight(i, j)}{\sum_{jj=1}^{DiscK} weight(i, jj)} (2). \quad (35)$$

This cost function allows to assign a superior weight to solutions that are of our interest (close-to-strike).

6. NUMERICAL RESULTS

6.1. Numerical results: uniform knots interpolation case

The result of the calibration is a local volatility surface introduced on Figure 6 and Figure 7 (same figure / different view). Market prices simulated with $\sigma = 0.2$ for all K, T are represented by the purple plane surface. Calibrated local volatility surface oscillates around it. From the Figure 7 we can conclude that oscillation is less sharp around $K = S = 100$ while for far-from-the-money options it becomes even negative.

The surface represents the local volatility matrix $\sigma(K, T)$. We studied the frequency of appearance of σ in different intervals. On Figure 8 we describe distribution of $\sigma(K, T)$ from the whole matrix (i.e. for all K, T). The histogram is obtained using mutation rate $m = 0.07$. We can see that most of the values lay in the interval $[0; 0.1]$ and $[0.3; 0.5]$.

Then the analysis is narrowed to the values of $\sigma(K, T)$ for all K_j , where index $j \in [20; 30]$, while at-the-money value is represented by index $j = 25$. Looking at only close-to-the-money options we notice that most of the values of σ lay in the interval $[0.1; 0.4]$, as we can see at Figure 9.

Quality criteria for these results is presented on Figure 10.

Three experiment sets are introduced: each set is executed with different mutation rate parameter and contains three similar experiments. In terms on number of iterations and estimated quality measure the mutation rate $m = 0.07$ turns out to be the best to reconstruct the local volatility surface.

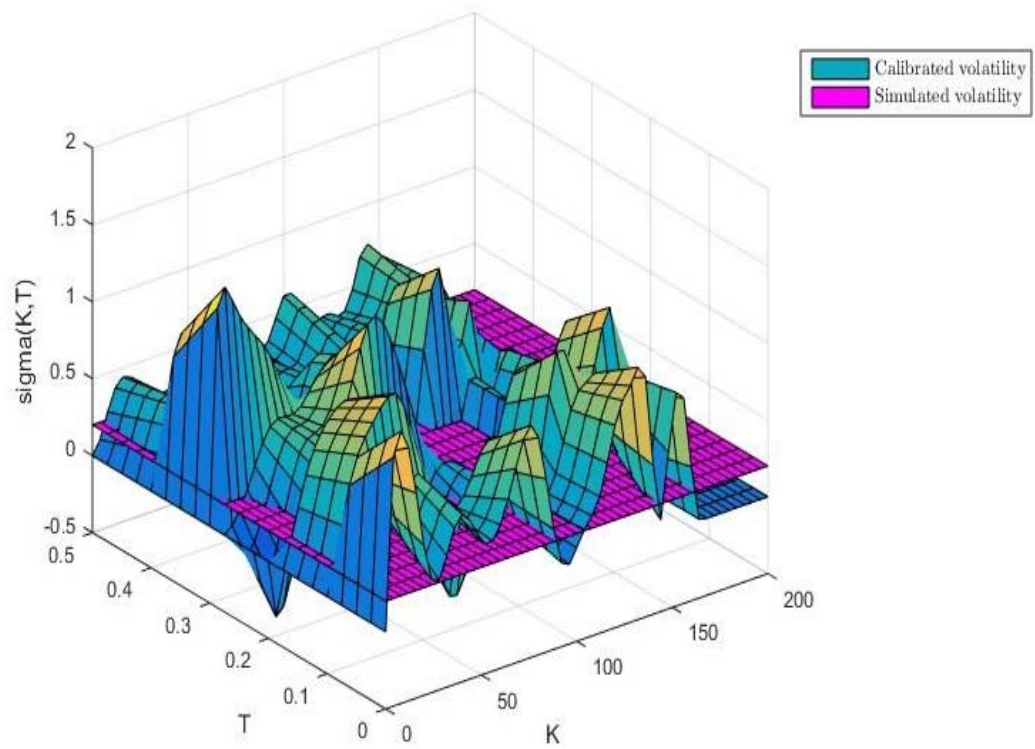


Figure 6. Local volatility surface

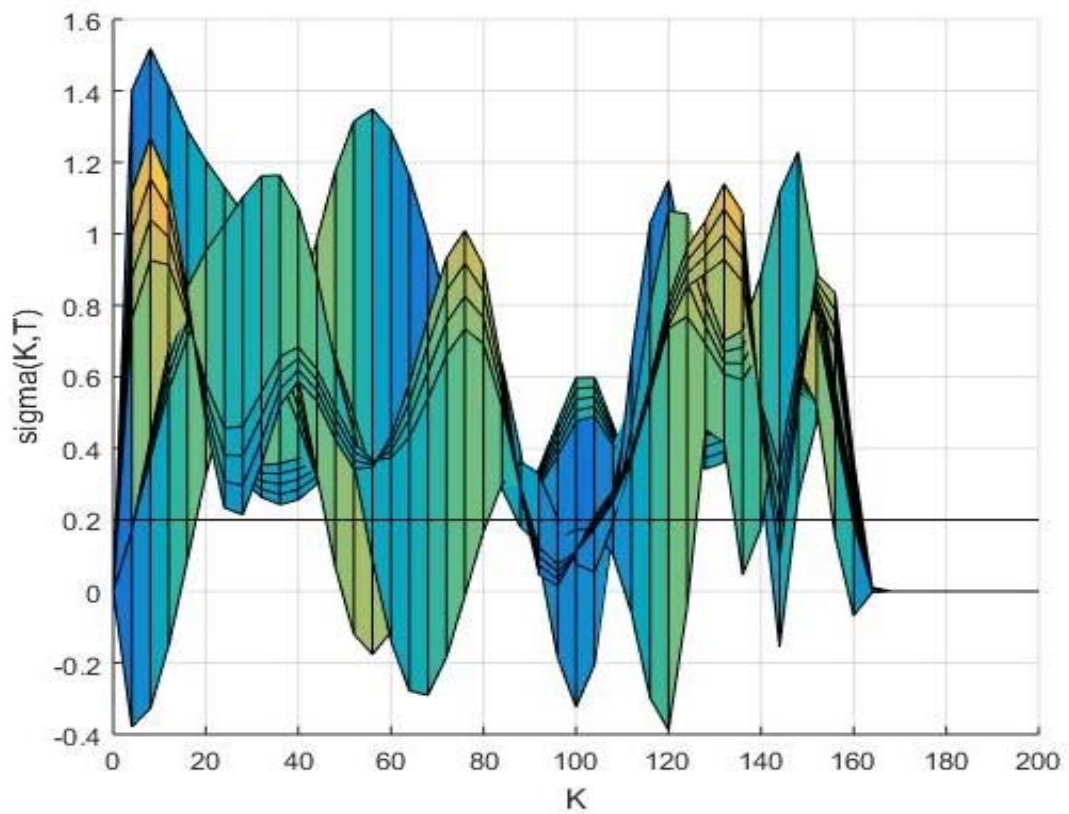


Figure 7. Local volatility surface: view K, σ

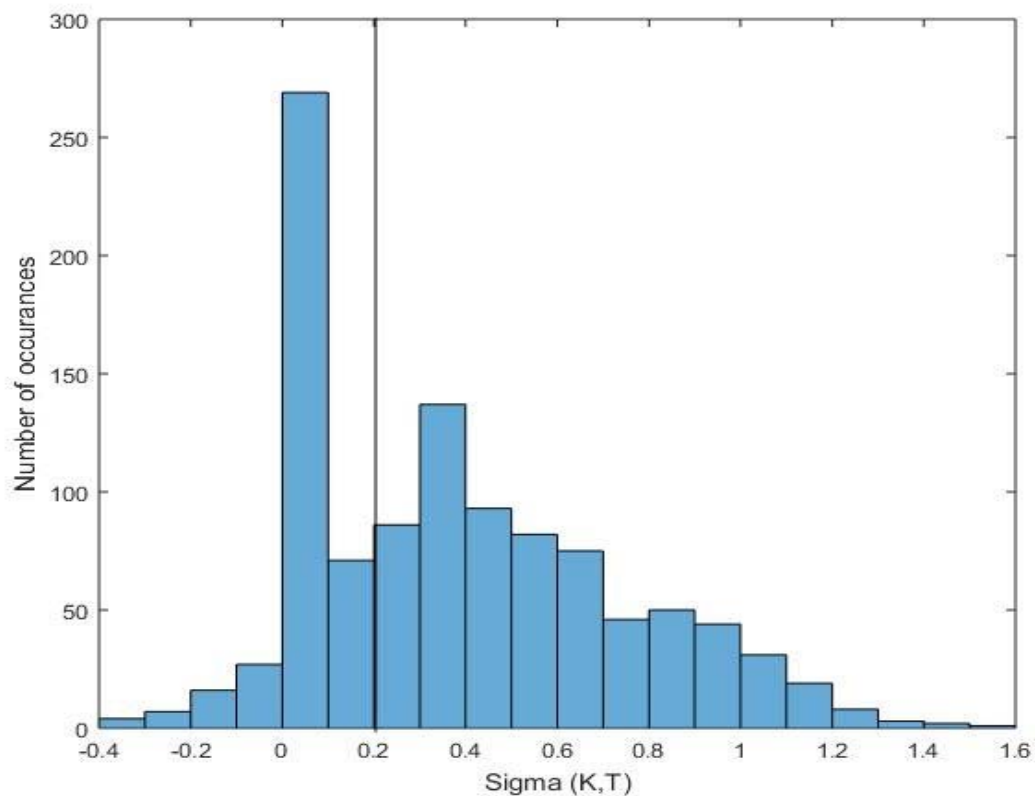


Figure 8. Local volatility surface: Histogram over all values

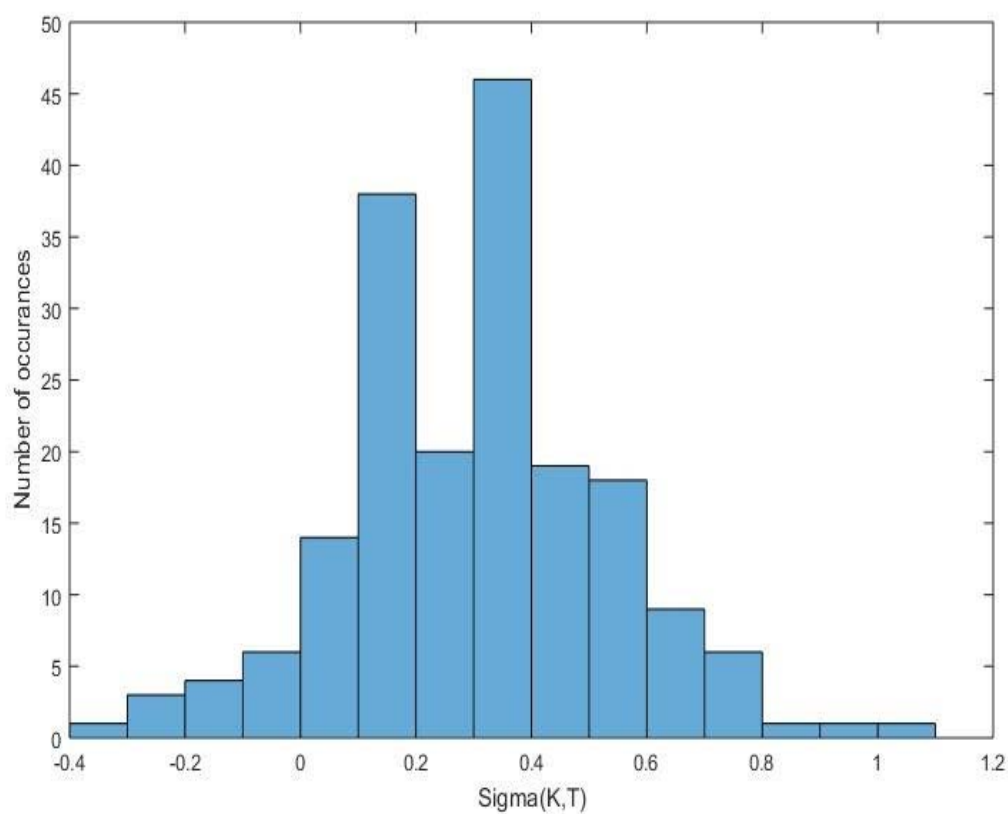


Figure 9. Local volatility surface: Histogram over index of $K \in [20; 30]$

| Experiment set | Experiment # | Mutation rate | Number of iterations | Quality measure (Q) [0,1] |
|----------------|--------------|---------------|----------------------|---------------------------|
| 1 | 1 | 0.05 | 677 | 0.545455 |
| 1 | 2 | 0.05 | 63 | 0.545455 |
| 1 | 3 | 0.05 | 309 | 0.545455 |
| 2 | 1 | 0.03 | 512 | 0.363636 |
| 2 | 2 | 0.03 | 792 | 0.545455 |
| 2 | 3 | 0.03 | 211 | 0.454545 |
| 3 | 1 | 0.07 | 538 | 0.636364 |
| 3 | 2 | 0.07 | 89 | 0.545455 |
| 3 | 3 | 0.07 | 281 | 0.545455 |

Figure 10. Results of the calibration

CONCLUSIONS

In current paper we have introduced the state of the art in local volatility calibration. It is based on results of Ben Hamida, Cont, Dupire, Del Moral, Cerf, articles and other research contributions. The genetic algorithm can be used as an alternative to gradient descent algorithm with smoothness requirement (problem regularization). Genetic approach allows to reflect the multiplicity of solutions (as we cope with an ill-posed problem) rather than get rid of it by regularization.

B-splines and linear interpolation are used to represent local volatility surface on strike and maturity axis respectively.

The input of genetic algorithm is a set of parameters of B-splines. Instead of optimizing directly the values of $\sigma(K, T)$ we use these parameters that also allows us to reduce the dimensionality of the problem. These parameters θ are then passed to Dupire PDE pricer that returns matrix of option values for all K, T (strikes and maturities). This matrix is used to evaluate the cost function by comparing these prices to simulated market prices. The set of B-spline parameters corresponding to accepted solution allows us to reproduce local volatility surface at desired points by interpolation.

We have implemented the algorithm and obtained coherent results. Our algorithm is able to reconstruct efficiently the local volatility surface starting from random values of control points in B-spline representation of local volatility so basically we are not obliged to use a prior (suggested by Ben-Hamida and Cont) to have the algorithm converged (but it can improve the convergence).

The value $m = 0.07$ turns out to be the most efficient mutation rate (Figure 10). It reduces at most the number of iterations while increasing algorithm's quality.

The quality measure estimating the percentage of solutions laying in 30% threshold from realized values is between 36% and 63% (Figure 10). We also calculated that for 60% threshold this percentage reaches 90%. The quality measure however takes into account only close-to-strike options (Figure 9). If we take the whole grid we can see from (Figure 8) that much less results lay in this interval. However, in practice we are not interested in this purely theoretical conclusion as volatility value does not affect at all the prices of far-from-strike options.

It means that the algorithm is capable of reconstructing coherent volatility surface that can be used in future research. At the same time, it reflects the multiplicity of solutions many of which satisfy quality criteria. However, to improve the quality we have to bring some additional modifications.

PROPOSITIONS

In further work we propose to extend the research of Ben-Hamida and Cont by using of De Boor algorithm of interpolation of parameters grid to obtain better convergence. The reason to try De Boor approach is that it allows us to apply genetic operators to control points that are not uniform on strike axis. We will be able to apply it to control points laying in a specific close-to-strike range. It is supposed to improve the convergence of the algorithm as far-for-strike solutions have very small weight in cost function.

We will also introduce different quality criteria that can include a measure of the quality of local volatility surface based on inter-strike and inter-maturity relations. Logically for the neighboring values of maturities or strikes (or both) the values of calibrated local volatilities should not differ a lot except at-the-money cases.

Generating of initial population from prior distribution proposed by Ben-Hamida and Cont can increase the speed of convergence. According to the authors the implied volatility of in-the-money options can be a good initial guess for the entire population.

These and other techniques including usage of different discretization scheme, weight function, interpolation order and sigma's positivity constraint (currently we obtain $\sigma < 0$ in far-from-strike cases) can significantly improve the convergence as well as improve quality measures obtained.

Obtained surfaces are used to produce a forecast of the "true" volatility surface. Such predictive analysis can be conducted using supervised and non-supervised machine learning techniques.

REFERENCES

1. Ben Hamida, S., & Cont, R. (2013). Recovering volatility from option prices by evolutionary optimization. *Journal of computational finance*, 8(4), 1-45. Retrieved from <http://www.proba.jussieu.fr/pageperso/ramacont/papers/evolution.pdf>
2. Bonnans, J. F., Cognet, J. M., & Volle, S. (2002). *Estimation de la volatilité locale d'actifs financiers par une méthode d'inversion numérique* (rapport de recherche No. 4648). Retrieved from <https://hal.inria.fr/inria-00071937/document>
3. Cerf, R. (1998). Asymptotic convergence of genetic algorithms. *Advances in Applied Probability*, 30(2), 521-550. <https://doi.org/10.1239/aap/1035228082>
4. Coleman, T. F., Li, Y., & Verma, A. (1999). Reconstructing the unknown local volatility function. *Journal of computational finance*, 2(3), 77-102. <http://dx.doi.org/10.21314/JCF.1999.027>
5. Del Moral, P., & Miclo, L. (2001). Asymptotic results for genetic algorithms with applications to non-linear estimation. In Kallel, L., Naudts, B., & Rogers, A. (Eds.), *Theoretical aspects of evolutionary computing* (439-493 pp.). Berlin: Springer-Verlag.
6. Dieterle, F. (2003). Variable selection by genetic algorithms. In *Multianalyte quantifications by means of integration of artificial neural networks, genetic algorithms and chemometrics for time-resolved analytical data* (Ph.D. Thesis). Tübingen: University of Tübingen. Retrieved from http://www.frank-dieterle.de/phd/2_8_5.html
7. Dupire, B. (1994). Pricing with a smile. *Risk*, 7(1), 18-20.
8. Geraghty, J., & Mohd Razali, N. (2011). Genetic algorithms performance between different selection strategy in solving TSP. *Proceedings of the World Congress on Engineering* (London, U.K.). Retrieved from https://www.researchgate.net/figure/236179246_Genetic_Algorithms_Performance_Between_Different_Selection_Strategy_in_Solving_TSP
9. Jackson, N., Süli, E., & Howison, S. (1998). Computation of deterministic volatility surfaces. *Journal of computational finance*, 2(2), 5-32. <http://dx.doi.org/10.21314/JCF.1998.022>
10. Lagnado, R., & Osher, S. (1997). A technique for calibrating derivative security pricing models: numerical solution of an inverse problem. *Journal of computational finance*, 1(1), 14-25. <http://dx.doi.org/10.21314/JCF.1997.002>