UNIVERSITY OF NAPLES FEDERICO II
Department of Economics and Statistical Sciences

PH.D. PROGRAM IN ECONOMICS
XXXIII CYCLE

Meshless Methods for Option Pricing and Risks Computation

Author:
FEDERICA SICA

Supervisor:
PROFESSOR EMILIA DI LORENZO

Co-Supervisor:
PROFESSOR SALVATORE CUOMO

"And although I have seen nothing but black crows in my life, it doesn't mean that there's no such thing as a white crow. Both for a philosopher and for a scientist it can be important not to reject the possibility of finding a white crow. You might almost say that hunting for 'the white crow' is science's principal task."

Jostein Gaarder, *Sophie's World*

"I will not go so far as to say that to construct a history of thought without profound study of the mathematical ideas of successive epochs is like omitting Hamlet from the play which is named after him. That would be claiming too much. But it is certainly analogous to cutting out the part of Ophelia. This simile is singularly exact. For Ophelia is quite essential to the play, she is very charming - and a little mad. "

Alfred North Whitehead

UNIVERSITY OF NAPLES FEDERICO II

Abstract

Faculty of Economics
Department of Economics and Statistical Sciences

Doctor of Philosophy in Economics

Meshless Methods for Option Pricing and Risks Computation

by FEDERICA SICA

In this thesis we price several financial derivatives by means of radial basis functions. Our main contribution consists in extending the usage of said numerical methods to the pricing of more complex derivatives - such as American and basket options with barriers - and in computing the associated risks. First, we derive the mathematical expressions for the prices and the Greeks of given options; next, we implement the corresponding numerical algorithm in MATLAB and calculate the results. We compare our results to the most common techniques applied in practice such as Finite Differences and Monte Carlo methods. We mostly use real data as input for our examples. We conclude radial basis functions offer a valid alternative to current pricing methods, especially because of the efficiency deriving from the free, direct calculation of risks during the pricing process. Eventually, we provide suggestions for future research by applying radial basis function for an implied volatility surface reconstruction.

Summary

In the last decades, financial engineers and firms have emphasized the importance of investigating alternative models and numerical methods to solve well-known problems in Mathematical Finance. Through mathematical models and their numerical implementation, it is indeed possible to price certain financial products, and derive the quantities which represent the risks of keeping such products in a given portfolio. By challenging the hypotheses of simpler models - such as a classical Black-Scholes framework [4] - prices and risks can be calculated more accurately, which translates in a profit from the deal, or in the reassurance the risks the firm will face are not higher than a predefined threshold. More complex models [15] [27] may better capture the uncertainty of the market drivers, allowing the portfolio manager to have a better overview of her positions.

The *price* of a product is usually derived as the solution of a partial differential equation associated to a given model. On the other hand, risks - or *Greeks* - are defined as the mathematical derivative or sensitivity of the price to a change in the inputs of the models. It can be argued that a *correct* and unique price does not exist, essentially because the models we deal with assume hypotheses which are not satisfied in the real world. For example, in the real market we will find a *bid* and an *ask* price for the same product, which correspond to a price respectively offered by the buyer and by the seller: in case of illiquidity of the product, such prices will hardly converge. On the other hand, if the product is continuously traded on the same market, bid and ask price will get closer and closer. There are mathematical theories which formalize a market with more than one price per product - e.g. conic Finance [35] - in which they distantiate themselves from a unique, risk neutral price and allow for a certain amount of risk, seen as acceptable by the portfolio manager. Classical models are derived in a risk neutral world and assume a complete market: this ensures that the price of an asset is equal to the discounted expected value of the future payoff under the associated, unique risk-neutral measure. Since we solve the equations derived from such models, we basically assume the resulting price is *correct*, and all numerical methods converge to the same price. Indeed, as George Box stated [5]: *'Essentially, all models are wrong, but some are useful.'*

Since not all the proposed mathematical models in literature exhibit an analytical solution - id est the price and/or risk can be computed via a single formula, it is necessary to rely on numerical methods. Monte Carlo methods [23] are the easiest to implement and adapt in case of high-dimensional problems, but do not provide the best achievable accuracy. Finite-Difference Methods (FDM) [13] prove to be the

best numerical technique in terms of accuracy when the given problem involves a low number of variables, but cannot always guarantee smooth risks profiles.
Meshless methods, in particular *Radial Basis Functions* (RBF) [17], are a valuable alternative numerical method to Monte Carlo and FDM. In this thesis we research the behavior of RBF in solving the partial differencial equations associated to models such as Black-Scholes or Heston. We derive the appropriate mathematical formulation in order to apply RBF framework and compute the solution which corresponds to a price. Furthermore, we derive the equations for the risks - or *Greeks* - associated to such price. We also explore eventual improvements to the numerical method through localization techniques such as Partition of Unity methods (PUM). Our main contribution relies in applying RBF with PUM to the pricing of more exotic financial products - such as barrier, American or Basket options - and in calculating the associated Greeks.
We compare the results with Monte Carlo and FDM, which represent the most known and used numerical methods by practitioners. The improvements achieved by RBF mainly consist in less computational effort in order to output the same results as FDM and Monte Carlo, and furthermore in the simplicity and elegance risks can be obtained with the new proposed methodology: resulting Greeks are even smoother and accurate than FDM-computed risks.
This thesis is structured as follows: we start with an introduction to financial models and products, then we show how to apply the above mentioned numerical techniques and eventually present our results. In the first chapter, we introduce the financial preliminaries required to understand the financial problems we address, explaining how introducing new hypotheses and model inputs helps better capture the market movements. We describe how to derive the associated partial differential equations and how to adapt them in order to price several financial products. In the second chapter we focus on a short description of Monte Carlo and FDM methods, before exploring RBF methods. We then move to the financial applications of such methods and show the numerical resolution associated to Black-Scholes and Heston models. In the third chapter, we present the corresponding results and investigate the advantages and the drawbacks of each method. We eventually provide our conclusions.
Most of the research presented in this thesis is covered by our work in:

- *Greeks computation in the option pricing problem by means of RBF-PU methods*, published in *Journal of Computational and Applied Mathematics*, 2020, by Sica Federica, Cuomo Salvatore and Toraldo Gerardo;

- *RBF methods in a Stochastic Volatility framework for Greeks computation*, published in *Journal of Computational and Applied Mathematics*, 2020, by Sica Federica, Cuomo Salvatore and Piccialli Francesco;

- *A Note on the Numerical Solution of Heston PDE*, published in *Ricerche di Matematica*, 2019, by Sica Federica, Cuomo Salvatore and Di Somma Vittorio;

- *Implied volatility surface reconstruction by means of radial basis functions*, working paper, by Sica Federica, Cuomo Salvatore, Alessandra De Rossi and Rizzo Luca.

We also refer to our work regarding Monte Carlo methods and the analysis of a financial Internet of Things system in:

- *Remarks on a financial inverse problem by means of Monte Carlo Methods*, published in *Journal of Physics Conference Series*, 2017, by Sica Federica, Cuomo Salvatore and Di Somma Vittorio;
- *Analysis of a data-flow in a financial IoT system*, published in *Procedia of Computer Science*, 2017, by Sica Federica, Cuomo Salvatore and Di Somma Vittorio;
- *An application of the one-factor Hull-White model in an IoT financial scenario*, published in *Sustainable Cities and Society*, 2018, by Sica Federica, Cuomo Salvatore and Di Somma Vittorio.

Contents

Dedicated to my grandparents:
Nonno Sabato and Nonna Sabina.

Chapter 1

Financial Preliminaries

Through our work, we aim to explore several numerical methods which can be applied to solve certain types of financial problems. We focus on pricing models, id est mathematical formulations which allow to calculate - analytically and non - the price of a financial product.
We will deal with typical instruments in the financial markets, such as interest rates or stocks. *Interest rates* represent the return - usually in percentage - of a cash investment in a given currency. For example, by investing an amount A of Euros today, the market tells us we will get an amount $(r_{ois} \cdot A + A)$ of Euros tomorrow, where r is today's overnight Euro interest rate expressed in percentage. If we prefer to invest the same amount A tomorrow, and get tomorrow's overnight rate in two days from now, we may choose to fix the future overnight rate today: such rates are called *forward rates*. Rates are implied from the market expectations of the market participants, and therefore reflect what the market players believe will be a fair rate for an investment made at a certain time t, with expected return in a day, months or years.
Different from rates but still important bricks of several, more complex financial products, are *stocks* or, equivalently, *equities*. Investing in Apple's or Tesla's stocks means being entitled to a proportion of the company's assets and profits proportional to how much equity is owned. Stocks can be sold or bought in the market, and may also pay *dividends*, i.e. the part of profit that is due to the stocks' owner.
Financial derivatives are products which base their value on an *underlying* asset, which may be an interest rate or an equity. A set of financial instruments and derivatives is called *portfolio*, and sums up all the information about the positions takes by a trader.
One of the first historical examples of the derivative we will price - *options* - is to be found in Aristotle's *'Politics'* or *'Politiká'*. Aristotle tells the tale of Thales $(624/623 - 548/545\ BC)$, a Greek philosopher and mathematician, who bought for a low price all the olive oil mills in Miletus: he had indeed predicted a good year for olives production, and was able to later rent the mills for a higher price. An option is a financial derivative, based on an underlying such as a stock, that allows - without obligation - the owner to sell or buy the underlying at a prefixed price in the future. By buying the mills, Thales had basically bought an option to rent them in the future at a fixed - and of course higher - price, making a profit.
We try to predict the future movements of financial instruments like options through

mathematical models: by looking at the price action of their underlyings in the market, we suppose they follow certain mathematical distributions, i.e. their price may move just like a normally or a lognormally distributed variable. Given the distributions of their underlyings, we can derive a Partial Differential Equation (PDE) which represents the movement of the price of the options that depends on them.
For example, rates - in all currencies - have been positive for a long time and therefore have been represented by lognormal distributions. However, the recent economical developments of countries in the Euro, US dollar (USD), Japanese Yen zones - and more - brought the corresponding rates to fall below zero, hence becoming negative. This suggests a normal distribution is more appropriate to capture their movements in the market. On the other hand, equities cannot have a negative value and a lognormal distribution shows to be still a relatively accurate hypothesis: the Black-Scholes model relies on it.
The derivation of the *Black-Scholes-Merton model*, appeared for the first time in 1973 [4], is perhaps he most famous result in Mathematical Finance. The classical model for option pricing, referred to as **Black-Scholes standard equation** (BS), is a linear parabolic partial differential equation:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{1.1}$$

which models the price $V(S,t)$ of a financial contract whose value depends on the time t and the price $S(t)$ of an underlying asset. σ is the *volatility* of the underlying, which represents how much the underlying is expected to vary in time. Among the adopted hypotheses, there are the assumptions of constant volatility σ and risk-free interest rate r that appear in the equation: in this special case, an analytical solution for (1.1) is available. The solution matches the market price of **European options** on a single underlying S, i.e. one of the simplest derivatives to price.
For more exotic types of options, original hypotheses for equation (1.1) must be enriched resulting in a more complex partial differential equation with parameters depending on time and on the underlying S. For example, if the price V depends on more than one underlying S, equation (1.1) becomes a multi-dimensional equation. Models that assume the volatility to be a process $\sigma(S(t),t)$ are called **local volatility models**; models with a stochastic volatility process are referred to as **stochastic volatility models**. The need for these extensions derives from observations of the market implied volatility: plotting the volatility against the option's maturity or strike shows a skew that is not foreseen by the classical Black-Scholes equation. Thus, a stochastic or local volatility model will provide a more accurate estimation of the price.
For extended versions of the classical Black-Scholes PDE, analytical solutions are not any more available. Numerical techniques are required to compute an accurate solution, i.e the price of the derivative V. Furthermore, traders are interested in the

sensitivity of the price V to its parameters: S, σ, r and T. Such quantities have a significant impact on the portfolio management and are generally referred to as *Greeks*, since they are indicated by Greek alphabet's letters. Greeks are basically the derivatives of a certain order of the price V with respect to its parameters.
The purpose of this thesis consists in exploring the adaptation of new numerical methods to existing financial models in order to efficiently price several types of options and compute the corresponding Greeks.
In this chapter we introduce the financial concepts and formulae which are relevant for this research. We present the classical Black-Scholes framework and its extensions respectively in sections (1.1) and (1.2). Additionally, in section (1.3) we describe the financial products for which we will derive and calculate the price. We eventually explain the importance of Greeks, i.e. the sensitivities of the price to the model's factors, in section (1.4).

1.1 Classical framework

The **Black-Scholes-Merton** model was originally conceived in 1969, but it was later published with its derivation in 1973 [4]. The model consists in a linear parabolic partial differential equation which defines the behavior of a financial product, whose value $V(S(t), t)$ depends on an asset $S(t)$ and, naturally, on time t. We will refer to $S(t)$ as the *underlying*, and it may represent a stock, a forward rate or another asset price.

1.1.1 Black-Scholes derivation

We will now heuristically derive the Black-Scholes equation. Let us consider the value of a portfolio Π which we build as following:

- We buy a derivative $V(S(t), t)$. We say we have a *long position* in $V(S(t), t)$.
- We sell a quantity Δ of the underlying $S(t)$. We say we have a *short position* in $S(t)$.

Therefore, the value of the portfolio Π will be:

$$\Pi = V(S(t), t) - \Delta S(t). \tag{1.2}$$

The first hypothesis of Black-Scholes-Merton model consists in assuming the underlying $S(t)$ is lognormally distributed. Formally:

$$dS = \mu S dt + \sigma S dW, \tag{1.3}$$

where μ is the *drift*, σ the *volatility* of the asset and W a Brownian motion. In the original BS framework, such parameters are simply constants. $S(t)$ is therefore a Geometric Brownian motion (GBM).

We can now calculate the change of the portfolio Π from time t to time $t + dt$, where dt stands for an infinitesimal change in time:

$$d\Pi = dV(S(t), t) - \Delta dS(t). \tag{1.4}$$

By applying Itô's formula [48], we obtain $dV(S(t), t)$:

$$dV(S(t), t) = \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS(t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt. \tag{1.5}$$

Substituting (2.15) in equation (2.10), we have:

$$\begin{aligned} d\Pi &= \frac{\partial V}{\partial t}dt + \frac{\partial V}{\partial S}dS(t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt - \Delta dS(t) \\ &= \frac{\partial V}{\partial t}dt + \left(\frac{\partial V}{\partial S} - \Delta\right)dS(t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt. \end{aligned} \tag{1.6}$$

To obtain a risk-free portfolio, we would like to get rid of the **randomness** factors, which means we need to cancel out the terms in dS and leave only the deterministic terms in dt. The easiest way to reach our goal is to choose:

$$\Delta := \frac{\partial V}{\partial S}. \tag{1.7}$$

The process of eliminating risk deriving from the underlying $S(t)$ is called **delta hedging**. Hence, if at each time step we choose a quantity Δ as defined, we will obtain a deterministic portfolio Π:

$$d\Pi = \frac{\partial V}{\partial t}dt + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt, \tag{1.8}$$

consisting in only deterministic terms.
In order to continue with the derivation of the BS equation, we require another important assumption: the **no arbitrage** principle. Indeed, in an arbitrage-free world the deterministic portfolio Π must benefit of the same return of a riskless portfolio:

$$d\Pi = r\Pi dt, \tag{1.9}$$

where r is the risk-free rate of the market. Imposing such equality and plugging in (2.38) in (1.8) holds:

$$r\Pi dt = \frac{\partial V}{\partial t}dt + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt, \tag{1.10}$$

and substituting (2.10) for Π:

$$r\Big(V(S(t), t) - \Delta S(t)\Big)dt = \frac{\partial V}{\partial t}dt + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt. \tag{1.11}$$

Considering that we chose a Δ quantity as defined in (1.7) and simplifying the expression above, we finally obtain the **Black-Scholes** partial differential equation

(PDE):

Black Scholes PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0. \tag{1.12}$$

It is interesting to note that the drift term, μ, of the underlying $S(t)$ does not appear as a parameter in the final equation. By considering that μ is the average return we expect from the underlying $S(t)$, it is clear why it is dropped: we cannot expect any other return than the risk-free rate from a riskless, deterministic portfolio.

1.1.2 Black-Scholes solution

Most of the financial partial differential equations are of the type of (1.12): they are linear, meaning that the sum of two solutions is still a solution, and parabolic, i.e. they can be reduced to a heat equation [16]. In particular, even if there exists some discontinuity in the terminal conditions, the solution will always be smooth. Furthermore, if the solution does not grow too fast with the underlying $S(t)$, the solution will also be unique.
The great advantage of the BS model consists in providing an exact solution. Exact solutions are quite rare in Finance and applied sciences in general. The analytical expression can be derived by means of different methods: by transforming the BS equation into a basic diffusion equation, by Fourier or Laplace transforms, by using Green's function or Fourier series.
The particular BS solution depends on the type of contract $V(S(t), t)$ we aim to price. Each contract yields a different terminal condition which will impact the analytical expression of the corresponding solution. Furthermore, it is important to notice the BS model allows for analytical formulae for the sensitivities of the solution to the main risk-factors. We will provide the BS solution and sensitivities for several contracts in sections (1.3) and (1.4).

1.2 Black-Scholes extensions

First we summarize and briefly discuss the main hypotheses under which the classical BS framework is valid and analyze the limits of each assumption:

- HP 1. The lognormality of the underlying $S(t)$. In particular, we assumed a constant volatility. This hypothesis is too simplistic, and does not allow for a thorough description of the market dynamics.

- HP 2. There are no *transaction costs* and no *arbitrage opportunities*. Hence, we assume the market is complete and arbitrage-free. This restriction is clearly valid only in theory: there are always transactions costs involved and, moreover, in the real world the same cashflows may have a different price, consequently causing arbitrage opportunities from which market makers will profit.

- HP 3. The underlying $S(t)$ pays no dividends. This restriction can be easily dropped by slightly extending the model.
- HP 4. The constant or time-dependent risk-free rate is known in advance. Furthermore, the borrowing and landing rates coincide. In general, interest rates are stochastic and therefore unknown: the model can be extended to the case of *stochastic* rates.
- HP 5. *Delta hedging* is continuously done. In practice, this would mean we continuously buy and sell a Δ quantity of the underlying $S(t)$ in order to keep a risk-neutral portfolio. This is not true in the real world, since transactions can be done only in discrete time.

Clearly, the need to overcome such limitations resulted in several extensions of the original BS model.

1.2.1 Dividend paying assets

The BS framework can be easily extended to take into account dividend paying assets [42]. Indeed, if we assume the asset $S(t)$ pays a continuous, constant dividend return D, then at each dt it is paid the following quantity:

$$DS(t)dt. \tag{1.13}$$

This translates in an extra deterministic factor $-D\Delta Sdt$ in equation (2.18):

$$d\Pi = \frac{\partial V}{\partial t}dt + \left(\frac{\partial V}{\partial S} - \Delta\right)dS(t) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}dt - D\Delta Sdt. \tag{1.14}$$

After similar manipulations to the equations in section (1.1), we obtain the modified **Black-Scholes equation for dividend paying assets**:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r - D)S\frac{\partial V}{\partial S} - rV = 0. \tag{1.15}$$

The corresponding analytical solution is similar to the original Black Scholes formula, and can be obtained by simply exchanging the term r with $r - D$.

1.2.2 Time-dependent parameters

Another simple extension consists in considering deterministic, time-dependent risk-free rate r, drift μ and volatility σ of the underlying asset.
We assume the following dynamics for the risk-free rate:

$$d\Pi(t) = r(t)\Pi(t)dt, \tag{1.16}$$

which naturally yields:

$$\Pi(t) = e^{\int_0^t r(s)ds}. \tag{1.17}$$

Secondly, we keep the lognormal dynamics for the asset $S(t)$, and introduce time-dependent drift and volatility. The solution to the stochastic differential equation (SDE) (1.3) becomes:

$$S(t) = S(0)e^{\int_0^t \sigma(s)dW(s) + \int_0^t (\mu(s) - \frac{\sigma^2(s)}{2})ds}. \tag{1.18}$$

Following section (1.1), we obtain the same BS equation with time-dependent parameters. The analytical solution will show integrals over time of the risk-free rate and the volatility, instead of their corresponding constant values.

1.2.3 Multi-dimensional problem

In case that the derivative $V(S(t), t)$ depends on multiple underlyings $S_0(t)$, $S_1(t)$, $\ldots$, $S_n(t)$. We shall consider the dynamics for each asset $S_i(t)$:

$$dS_i = \mu_i S_i dt + \sigma_i S_i dW_i, \qquad i = 0, \ldots, n \tag{1.19}$$

where μ_i, σ_i and W_i are the drift, the volatility and the Brownian motion associated to the asset $S_i(t)$, respectively. Each Brownian motion W_i is correlated with the other assets' Brownian motions by the correlation coefficients ρ_{ij}:

$$E[dW_i dW_j] = \rho_{ij} dt. \tag{1.20}$$

We define the **correlation matrix** Σ as:

$$\Sigma = \begin{bmatrix} 1 & \ldots & \rho_{1n} \\ \vdots & \ddots & \\ \rho_{n1} & & 1 \end{bmatrix}$$

with entries $\rho_{ii} = 1$ on the diagonal and $\rho_{ij} = \rho_{ji}$. Correlations are usually estimated from time series or implied from the market.
Using the multidimensional Itô's lemma [48], we obtain the **multi-dimensional Black-Scholes**:

Multi-dimensional Black Scholes PDE

$$\frac{\partial V}{\partial t} - \frac{1}{2}\sum_{i,j=1}^{n} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i=1}^{n} r S_i \frac{\partial V}{\partial S_i} - rV = 0. \tag{1.21}$$

Following the reasoning established in section (1.2.1), we obtain the multi-dimensional Black-Scholes for dividend-paying assets:

$$\frac{\partial V}{\partial t} - \frac{1}{2}\sum_{i,j=1}^{n} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i=1}^{n} (r - D_i) S_i \frac{\partial V}{\partial S_i} - rV = 0. \tag{1.22}$$

1.2.4 Volatility smile

BS model assumes a constant or time-dependent volatility. Since the model also provides an analytical formula for the price of simple derivatives, it is natural to try to derive the volatility of the underlying $S(t)$ by inverting the given formula. Such volatility is called **implied volatility**, and can be interpreted as the expected value of the future volatility by market participants. By simply plotting the values corresponding to the same type of contract with various parameters, traders observed the well-known shape of a **volatility smile**: volatility is actually not constant, but varies with the derivative's $V(S(t), t)$ parameters.
To capture the volatility smile, researchers have enriched the BlackâĂŞScholes model by modeling the volatility as a function of both underlying asset and time: the most famous attempt is the **local volatility model** by Dupire, Derman and Kani ([15], [10], [14]). Such model is easy to calibrate and guarantees the completeness of the market. A further extension consists in considering a **stochastic volatility**, which exhibits an intrinsic source of randomness given by a Brownian motion. The **Heston model** [27] is a stochastic volatility model which considers a more realistic asset distribution than the original, lognormal distribution of the BS framework. Furthermore, it provides an analytical solution for European options: this result is extremely useful for the calibration of the model itself.

The dynamics introduced by Heston for the underlying asset $S(t)$ are:

Heston model dynamics

$$dS(t) = rS(t)dt + \sqrt{\nu(t)}S(t)dW_1(t) \tag{1.23}$$

$$d\nu(t) = k[\theta - \nu(t)]dt + \sigma\sqrt{\nu}dW_2(t) \tag{1.24}$$

$$dW_1(t)dW_2(t) = \rho dt \tag{1.25}$$

where we have defined:

- r is the constant risk-free interest rate;
- ν is the volatility of the underlying;
- σ is the *volvol* or volatility of volatility;
- k is the mean reversion speed of the volatility;
- θ is the mean reversion level;
- $W_1(t)$ and $W_2(t)$ are two correlated Wiener processes with correlation ρ.

It can be shown that the value $V(S, t)$ of a derivative with underlying $S(t)$ that follows Heston model's dynamics satisfies the following **Heston PDE**:

Heston PDE

$$\frac{\partial V}{\partial t} + \frac{1}{2}\nu S^2 \frac{\partial^2 V}{\partial S^2} + \rho\sigma\nu S \frac{\partial^2 V}{\partial S \partial \nu} + \frac{1}{2}\sigma^2 \nu \frac{\partial^2 V}{\partial \nu^2} + rS\frac{\partial V}{\partial S} + k(\theta - \nu)\frac{\partial V}{\partial \nu} - rV = 0. \quad (1.26)$$

1.2.5 Continuous hedging and transaction costs

Among the other BS hypotheses, it is assumed the continuity of delta hedging. In practice, it is not only impossible to continuously hedge, but it could also be too expensive due to transaction costs. Boyle and Emanuel [6] studied the behavior of continuous and discrete hedging: they considered a small timestep $\delta t = 0.01$ for the first case and a larger time interval for the discrete case. They found the hedging error is a random variable with a chi-squared distribution, which can have an impact on the total profit and loss (P&L) of the portfolio.

BS also assumes that no transaction costs take place when hedging: in equity and emerging markets derivatives this is definetely not true because of illiquidity. In 1985, Leland [34] proposed a model which allows for hedging at any timestep including cases in which the strategy would be sub-optimal. Leland formulates the following discrete dynamics for the underlying $S(t)$:

Leland model

$$\delta S = \mu S \delta t + \hat{\sigma} S dW, \quad (1.27)$$

where δt is a finite, constant and rebalancing interval, and $\hat{\sigma}$ is the augmented, hedging volatility defined as:

$$\hat{\sigma} = \sigma \sqrt{1 + \frac{k\frac{2}{\pi}}{\sigma\sqrt{\delta t}}} \quad (1.28)$$

where k is the proportional transaction costs rate - measured as a fraction of the value of transactions - and σ is the actual volatility of the asset $S(t)$. Hence, in this framework, transaction costs are proportional to the underlying value.
The model can be extended to an arbitrary payoff, leading to a non-linear parabolic partial equation, i.e. the value of a portfolio is not anymore the sum of the values of its components.

1.3 Overview of financial products

In this section we will provide the main definitions for the financial products we will price. One of the most common derivatives traded in the market are **options**. The idea behind this derivative is to have a sort of protection or gain against the rise or fall of the price of the underlying $S(t)$. An important quantity related to options is the payoff:

Definition 1.3.1. We define as **payoff** $\phi(S,t)$ the gain or loss function of an option on the underlying $S(t)$ at any time t of the option life.

In the following subsections, we will present various types of options, with their definition and boundary conditions. We will also show the analytical solution for their price, if any.

1.3.1 European options

Definition 1.3.2. A **European call** (respectively **put**) option on the amount of S units of currency, with **strike price** K and **exercise date** T, is a contract written at $t = 0$ with the following property: the holder of the contract has, **exactly at the time** $t = T$, the right but not the obligation to buy (respectively sell) S at the price K.

The adjective *European* defines the time at which the option can be exercised, i.e. the holder can decide to sell or buy the underlying at price K. The adjectives *call* or *put* simply define if the holder can buy or sell the underlying. The option expires on a date T, which denotes the *maturity* of the contract. After such date, the option cannot be exercised anymore and loses any value.
European options are also referred to as *vanilla*, meaning they are the simplest option contract to price. Indeed, we can use the analytical BS formula to find their price: in order to do so, we require additional information regarding the initial and boundary conditions of the Black-Scholes PDE (1.12).

Black-Scholes analytical formulae. Since the Black Scholes is typically solved backwards in time, we will refer to the initial condition as the *terminal condition* at time $t = T$.

Terminal Conditions for European Options

$$V(S,T) = \phi(S,T) := max(S(T) - K, 0) \qquad \text{for a call} \tag{1.29}$$

$$V(S,T) = \phi(S,T) := max(K - S(T), 0) \qquad \text{for a put.} \tag{1.30}$$

Therefore, the terminal condition is defined by the payoff function ϕ. The boundary conditions associated to the PDE for European options are:

Boundary Conditions for European Call Options

$$V(S,t) = 0 \quad \text{as} \quad S \to 0 \tag{1.31}$$

$$V(S,t) = S \quad \text{as} \quad S \to +\infty. \tag{1.32}$$

Boundary Conditions for European Put Options

$$V(S,t) = Ke^{-r(T-t)} \quad \text{as} \quad S \to 0 \tag{1.33}$$

$$V(S,t) = 0 \quad \text{as} \quad S \to +\infty. \tag{1.34}$$

Given terminal and boundary conditions, the Black-Scholes analytical formulae for European call and put options become:

Black Scholes formulae for European Options

$$V_{call}(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2) \tag{1.35}$$

$$V_{put}(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1) \tag{1.36}$$

where $N(\cdot)$ is the cumulative normal distribution and:

$$d_1 := \frac{\log\left(\frac{S}{K}\right) + (r + \frac{1}{2}\sigma^2(T-t))}{\sigma\sqrt{T-t}} \tag{1.37}$$

$$d_2 := d_1 - \sigma\sqrt{T-t} \tag{1.38}$$

We show in figure (1.1) the price for a European call option calculated via Black-Scholes formulae.

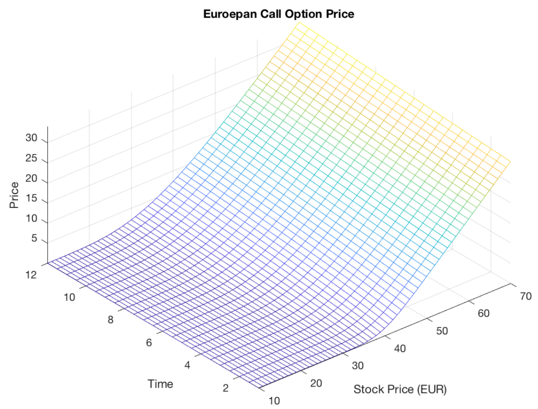


FIGURE 1.1: Price of a European call option calculated via Black-Scholes formulae, as function of time and stock price in EUR.

Heston analytical formulae. For a European option, the Heston model provides an analytical formula as well. For sake of simplicity, we will provide the formula for a European call option.

Let us consider a rectangular spatial domain of $[0, S^*] \times [0, \nu^*]$, where S^* and ν^* are truncated values of S and ν. Then, the natural boundary conditions for a European call option with maturity T and strike K are [3]:

Terminal and Boundary Conditions for European Call Options

$$V(S,\nu,T) = \max(S-K,0), \tag{1.39}$$

$$V(0,\nu,t) = 0, \tag{1.40}$$

$$\frac{\partial V(S^*,\nu,t)}{\partial S} = 1, \tag{1.41}$$

$$V(S,\nu^*,t) = S, \tag{1.42}$$

$$rS\frac{\partial V(S,0,t)}{\partial S} + k\theta\frac{\partial V(S,0,t)}{\partial \nu} - rV(S,0,t) + \frac{\partial V(S,0,t)}{\partial t} = 0. \tag{1.43}$$

Following Heston's paper [27], the analytical solution for the price of a vanilla European option is given by:

Heston formula for European Call Options

$$V(S,\nu,t) = SP_1 - Ke^{-r(T-t)}P_2 \tag{1.44}$$

where:

$$P_{k\in\{1,2\}}(\ln S,\nu,T;\phi) = \frac{1}{2} + \frac{1}{\pi}\int_0^{+\infty} Re\left[\frac{e^{-i\phi\ln K} f_k(\ln S,\nu,T;\phi)}{i\phi}\right]d\phi \tag{1.45}$$

with f_k the characteristic functions as described in [27].

1.3.2 American options

Definition 1.3.3. An **American call** (respectively **put**) option on the amount of S units of currency, with **strike price** K and **exercise date** T, is a contract written at $t=0$ with the following property: the holder of the contract has, **at any time** $t \leq T$, the right but not the obligation to buy (respectively sell) S at the price K.

Since in the American case the holder can exercise the option at any time, the value of this option will be greater than the respective European option with the same properties.

Penalty method formulation. An American option can be exercised at every time t: this implies a *free-boundary problem*. A common technique to solve free-boundary value problems is applying a *penalty method*. Basically, a *penalty term* is added to the equation to convert the problem into a fixed domain one. In this way we will obtain new and more feasible initial and boundary conditions ([29], [40]). The advantage of the penalty term formulation consists in using the same mathematical techniques for one or more underlyings, allowing the resolution of the PDE by means of any numerical method.

In order to apply a penalty method, we now want to formulate the *Linear Complementarity Problem* (LCP) for an American put option $V(S(t), t)$ [42]. Let us consider what happens when we are given the right to exercise the option at each time t. From a no-arbitrage argument, we easily obtain the constraint:

$$V(S,t) \geq \phi(S,t), \tag{1.46}$$

where we recall $\phi(S,t)$ to be the payoff of the option at time t. Indeed, if $V(S,t) < \phi(S,t)$, then we can easily achieve arbitrage by purchasing the option, exercising it, and eventually buying the underlying S in the market, realizing a risk free profit of $K - V - S$.

Therefore, we are left with two cases:

- $V(S,t) = \phi(S,t)$: it is more profitable to exercise the option. The equality in the BS PDE (1.12) is satisfied.
- $V(S,t) > \phi(S,t)$: it is more profitable to keep the option and maybe exercise it at a later time. The BS PDE (1.12) becomes an inequality:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \leq 0 \tag{1.47}$$

We can summarize such reasoning by formalizing it with the following:

LCP for American Options

$$\mathcal{L}V \geq 0 \tag{1.48}$$

$$V - \phi \geq 0 \tag{1.49}$$

$$(\mathcal{L}V = 0) \vee (V - \phi) = 0 \tag{1.50}$$

where we have defined the operator:

$$\mathcal{L} := \frac{\partial}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r. \tag{1.51}$$

We introduce a positive penalty parameter $\epsilon \in \mathcal{R}^+$ such that [29]:

Penalty method formulation for American Options

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV - \frac{max(\phi - V, 0)}{\epsilon} = 0. \tag{1.52}$$

Indeed, for $\epsilon \to 0$ we ensure that for a small $\delta \in (0,1)$ the quantity $\phi - V \leq \delta$. In practice, if $\phi - V \geq 0$, we re-obtain the Black-Scholes equation; in case $0 \leq \phi - V \leq \delta$, we impose an inequality as in (1.47).

1.3.3 Barrier options

Definition 1.3.4. A **Barrier option** is a European or American call or put option with the property of changing its value when the underlying price S touches a pre-defined barrier B. We can distinguish between:

- *knock-in*: the price of the underlying hits the barrier and the option comes into existence;
- *knock-out*: the price of the underlying hits the barrier and the option ceases to exist.

The payoff for a knock-out, call barrier option is:

$$\phi(S,t) = \begin{cases} max(S(t) - K, 0) & \text{if } S(t) < B \\ 0 & \text{if } S(t) \geq B \end{cases} \tag{1.53}$$

There exist no analytical formulae for barrier options. However, several numerical methods can be applied to estimate the price.

1.3.4 Basket options

Definition 1.3.5. A **Basket option** is a European or American call or put option on a basket of N underlyings S_1, S_2, $\ldots$, S_N: its value is based on the average of the underlyings' prices.

The payoff of a basket call option is typically:

$$\phi(S,t) = max(\frac{1}{N}\sum_{i=1}^{n} w_i S_i(t) - K, 0). \tag{1.54}$$

where w_i denote the weights to apply to each underlying S_i.

As for the barrier options case, there exist no analytical formulae for most of basket options. In general, semi-analytical formulae are available or, for more complicated payoffs, numerical methods remain the only viable option.

Penalty method formulation. In case of American basket options, we can derive the penalty method formulation similarly to section (1.3.2).
We define:

$$q := K - \sum_{i=1}^{n} w_i S_i(t), \tag{1.55}$$

for an American put option. We define the following penalty term for basket options:

$$P(V) = \frac{\epsilon(rK - \sum_{i=1}^{n} w_i d_i S_i)}{V + \epsilon - q} \tag{1.56}$$

where $\epsilon \in (0,1)$. Because of its formulation, the penalty term will be very large as we approach the payoff and negligible when far from it. This choice will imply an error of $\mathcal{O}(\epsilon)$. Hence, the penalty formulation for this case will be [40]

Penalty method formulation for American Basket Options

$$\frac{\partial V}{\partial t} - \frac{1}{2}\sum_{i,j=1}^{n} \sigma_i \sigma_j \rho_{ij} S_i S_j \frac{\partial^2 V}{\partial S_i \partial S_j} + \sum_{i=1}^{n} r S_i \frac{\partial V}{\partial S_i} - rV - P(V) = 0. \tag{1.57}$$

1.4 Introducing Greeks

A crucial numerical issue in derivatives pricing consists in determining the values of the so-called **Greeks**, or hedge ratios, for each product [30]. Greeks are defined as the sensitivities of the derivative price with respect to a certain risk factor: e.g. the underlying $S(t)$, the interest rates, the volatility, the time, etc. These quantities are extremely meaningful in the hedging of an option position: it is usually more important to get an accurate estimate of the Greeks while renouncing precision in the option's price itself. Greeks represent the possible future uncertainty of the portfolio and the trader's exposure to market movements. A correct calculation of Greeks means protecting the portfolio from expected future losses.
For European options and other simple contracts that are priced via Black-Scholes model, there exist analytical formulae for Greeks. In other cases, numerical methods are required to calculate such quantities. In the following subsections we will describe the most common Greeks and show an example for the simple case of a European call option.

1.4.1 Delta

Definition 1.4.1. We define the Greek **Delta** as the first derivative of the option price with respect to its underlying price:

$$\Delta := \frac{\partial V(S,t)}{\partial S(t)}. \tag{1.58}$$

Definition (1.4.1) can be extended to the value Π of a portfolio: the total delta of Π will simply be the sum of the deltas of its individual components. If the delta of a certain portfolio is zero, the portfolio is said to be **delta neutral**. Maintaining a delta neutral portfolio by continuously buying and selling a delta quantity of the underlying is called *delta hedging*. This technique allows the trader to eliminate the risk associated to any movement in the underlying.

Delta for a European Call Option. The delta for a European call option in the classical Black-Scholes framework is:

$$\Delta = \phi(d_1). \tag{1.59}$$

Therefore, Δ in such case is always a positive, bounded quantity:

$$0 < \Delta < 1. \tag{1.60}$$

The price of a European call option is therefore an increasing function of the underlying $S(t)$. Furthermore, we observe from (1.37) that:

$$\lim_{S(t)\to 0^+} d_1 = -\infty \tag{1.61}$$

$$\lim_{S(t)\to +\infty} d_1 = +\infty \tag{1.62}$$

hence, for the price of a European call option $V_C(S,t)$ and its delta Δ the following holds:

$$\lim_{S(t)\to 0^+} V_C(S,t) = 0 \tag{1.63}$$

$$\lim_{S(t)\to +\infty} V_C(S,t) = +\infty \tag{1.64}$$

$$\lim_{S(t)\to +\infty} \Delta = 0 \tag{1.65}$$

$$\lim_{S(t)\to +\infty} \Delta = 1. \tag{1.66}$$

We show in figure (1.2) the delta for a European call option.

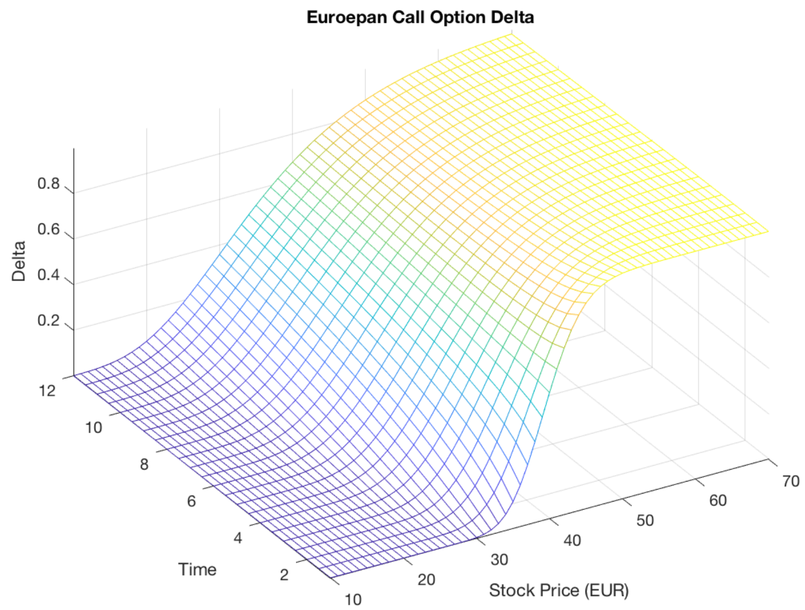


FIGURE 1.2: Delta of a European call option calculated via Black-Scholes formulae.

1.4.2 Gamma

Definition 1.4.2. We define the Greek **Gamma** as the second derivative of the option price with respect to its underlying price:

$$\Gamma := \frac{\partial^2 V(S,t)}{\partial S^2(t)}. \tag{1.67}$$

While the Δ of an option indicates how much the price of the option will move for a single unit of currency movement in the underlying price, the Γ determines how fast this movement will be: it measures the change in Δ with respect to the underlying. Γ gives information about how many times we have to hedge a position in order to keep it delta neutral: since hedging too often may be expensive, it is natural to wish for a low Gamma position.

Gamma for a European Call Option. The gamma for a European call option in the classical Black-Scholes framework is:

$$\Gamma = \frac{\phi'(d_1)}{\sigma S(t)\sqrt{T-t}}. \tag{1.68}$$

As for the Δ case, Γ is always a positive quantity. Hence, the Δ and the price V_C of a European call option are respectively an increasing and convex function of the underlying $S(t)$. Furthermore, we observe that:

$$\lim_{S(t)\to 0^+} \Gamma = 0 \tag{1.69}$$

$$\lim_{S(t)\to +\infty} \Gamma = 0. \tag{1.70}$$

We show in figure (1.3) the gamma for a European call option calculated via Black-Scholes formulae.

1.4.3 Theta

Definition 1.4.3. We define the Greek **Theta** as the first derivative of the option price with respect to time:

$$\Theta := \frac{\partial V(S,t)}{\partial t}. \tag{1.71}$$

Theta measures the change of the value of the option with respect to the passage of time: as the option approaches its maturity, its value will decrease. Because of this reason, Theta is also referred to as *time decay*.

Theta for a European Call Option. The theta for a European call option in the classical Black-Scholes framework is:

$$\Theta = -rKe^{-r(T-t)}\phi(d_2) - \frac{\sigma S(t)}{2\sqrt{T-t}}\phi'(d_1). \tag{1.72}$$

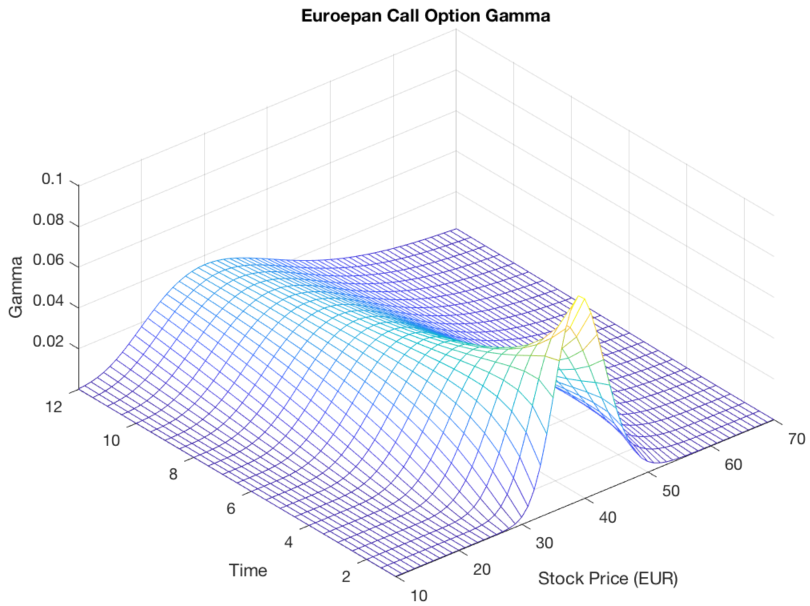


FIGURE 1.3: Gamma of a European call option calculated via Black-Scholes formulae.

We observe that theta is always negative for a call option and decreases with time. We show in figure (1.4) the theta for a European call option calculated via Black-Scholes formulae.

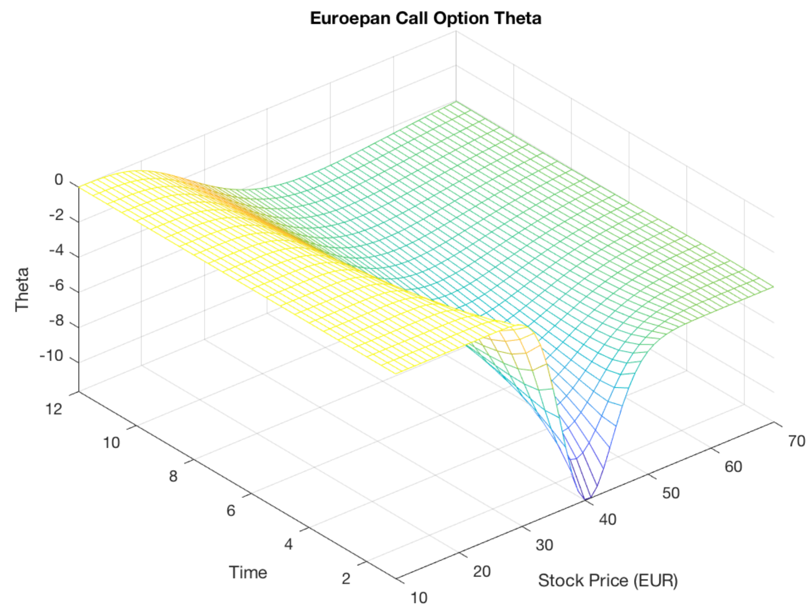


FIGURE 1.4: Theta of a European call option calculated via Black-Scholes formulae.

1.4.4 Vega

Definition 1.4.4. We define the Greek **Vega** as the first derivative of the option price with respect to volatility:

$$\nu := \frac{\partial V(S,t)}{\partial \sigma}. \tag{1.73}$$

Vega is usually indicated by the Greek letter ν. In general, the option price increases with volatility, since more uncertainty in the underlying increases the chances of a larger future payoff. Nevertheless, there are cases in which this is not true: for example, an option with a gamma that changes sign.

Vega for a European Call Option. The vega for a European call option in the classical Black-Scholes framework is:

$$\nu = S(t)\sqrt{T-t}\phi'(d_1). \tag{1.74}$$

We observe that vega is always positive for a call option and, in this case, the price V_C of the call option is strictly increasing with the volatility. Therefore, we may derive an inverse function of the price to calculate the *implied volatility*, i.e. the unique volatility value associated to the Black-Scholes price of the option. We show in figure (1.5) the vega for a European call option calculated via Black-Scholes formulae.

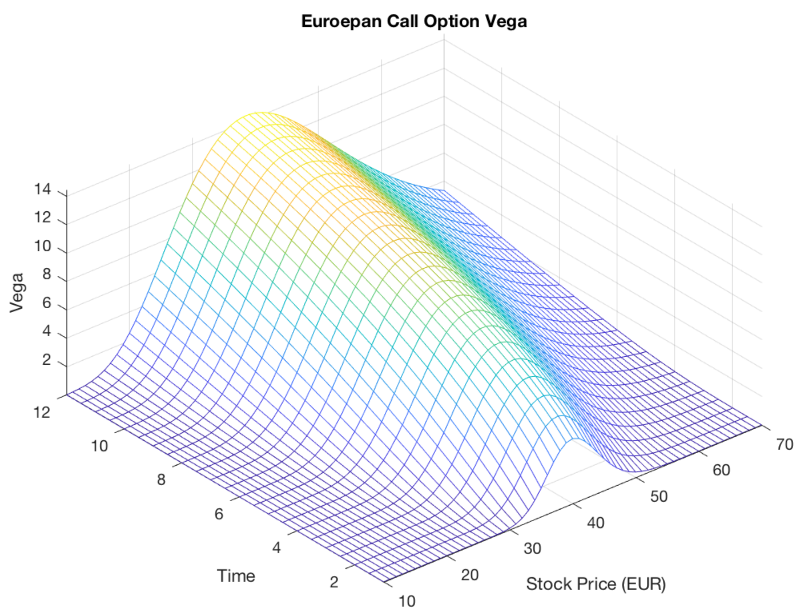


FIGURE 1.5: Vega of a European call option calculated via Black-Scholes formulae.

Chapter 2

Numerical methods for PDE solving

In this chapter we are going to show several numerical techniques which are required to compute an accurate solution of the proposed financial equations, i.e the price of the derivative V: the most common methods are Finite-Difference or Monte-Carlo based.

Finite-difference Metods (FDM) are more suitable for contracts which present an early-exercise feature or callable derivatives. For Greeks computation, FDM are the standard practice. On the other hand, they are usually harder to implement and slow in case of high-dimensionality problems. The initial and boundary conditions must be chosen in such a way to guarantee the stability of the solution.

Meshless methods have been recently proposed to solve the Black-Scholes PDE [49]. The advantages of methods such as Radial Basis Functions (RBF), together with localization techniques as Partition of Unity (PU), consist in obtaining a stable and accurate solution in a reasonable amount of time. Furthermore, they have a high potential as computational method for the Greeks' estimation: the method can be easily extended to calculate the Greeks values almost for free without affecting computational time. Our research focuses on meshfree methods: following the work in [49], we apply and extend meshless methods to solve financial PDEs: we discretize the spacial elliptic operator of the PDEs by means of RBFs and Partition of Unity method to calculate the price of vanilla and more exotic derivatives [53], [52], [50]. As a further novelty, we derive the formulae for the Greeks of the option under the RBF-PUM scheme.

Eventually, **Monte-Carlo** methods are the easiest to implement and more suitable for high-dimensional problems and some path-dependent derivatives. They consist in simulating the random walk and cashflows of the underlying S, calculating the average payoff and eventually returning its current value. It is slower than finite-difference methods up to four dimensions, but faster in case of more factors or underlyings. The error is of the order of the inverse square root of the number of simulations, i.e. an increase of 10% of the accuracy requires an increase by a factor of 100 in the number of simulations. In our work [51] we study an application to barrier option pricing, focusing on particle filtering techniques.

The chapter is organized as follows: in section (2.1) Finite-Difference methods are presented; in section (2.2) we describe meshfree methods and, eventually, section (2.1) is dedicated to Monte Carlo methods. Each section will contain a short description of the numerical method itself, together with its applications to solve financial PDEs.

2.1 Finite differences methods

Finite differences methods are the most common numerical resolution method for financial PDEs. Similarly to RBF, FDM solve a PDE by discretising both time and space derivatives as divided differences, and calculate the solution by iteratively applying the finite difference formulae at each time step.
As an example, let us consider a differentiable and real function $f : x \in \Re \longmapsto f(x) \in \Re$: we aim at finding appropriate approximations for its derivatives at some point $\bar{x}$. We indicate with h a small, positive number, which will denote the **mesh distance**. We may consider the following quantities to approximate the first derivative of f:

Forward difference formula

$$f'(\bar{x})_{fwd} = \frac{f(\bar{x}+h) - f(\bar{x})}{h}; \tag{2.1}$$

Backward difference formula

$$f'(\bar{x})_{bwd} = \frac{f(\bar{x}) - f(\bar{x}-h)}{h}; \tag{2.2}$$

Central difference formula

$$f'(\bar{x})_{cen} = \frac{f(\bar{x}+h) - f(\bar{x}-h)}{2h}. \tag{2.3}$$

For the second order derivative, we may consider the following finite-difference approximation:

$$f''(\bar{x}) = \frac{f(\bar{x}+h) - 2f(\bar{x}) + f(\bar{x}-h)}{h^2}. \tag{2.4}$$

Such formulae naturally derive from the Taylor expansion of the function f around the point $\bar{x}$. Clearly, the accuracy of the method is highly dependent on the continuity and differentiability order of the function f.
A finite-difference method may be *explicit* or *implicit*: if we are able to directly isolate and calculate at each step the value required for the next step, the method is explicit; otherwise, we may need to solve an implicit equation and use more time consuming root finding methods, such as Newton Raphson. In general, if we approximate the first derivative of an ordinary differential equation (ODE) via forward differences, the method will be explicit. If we involve backward or central difference formulae,

the method will likely be implicit. If an explicit method is simpler to implement, the implicit method will in general provide a better accuracy in the resolution.

2.1.1 FDM applications to Finance

We will provide three different and known approximations of the classical Black-Scholes equation via finite-difference methods: the Euler explicit method, the Euler implicit method, and eventually the Crank-Nicolson method. We consider a mesh of N equidistant space points for the underlying S and J equidistant time points: we indicate with $V_{n,j} := V(S_n, t_j)$ the price V calculated at the mesh point (n, j). We define the constant mesh distances $\Delta S := S_n - S_{n-1}$ and $\Delta t := t_j - t_{j-1}$. The boundary conditions will depend on the particular financial derivative we are pricing.

Explicit Euler method for Black-Scholes equation.
We discretize equation (1.12) by an explicit Euler method. The main difference with the other proposed FDM is the presence of the term $V_{n,j+1}$, at time point $j+1$, only on the left side of the equation.

$$\frac{V_{n,j+1} - V_{n,j}}{\Delta t} \approx \frac{1}{2}\sigma^2 S_n^2 \frac{V_{n+1,j} - 2V_{n,j} + V_{n-1,j}}{\Delta S^2} + rS_n \frac{V_{n+1,j} - V_{n-1,j}}{2\Delta S} + rV_{n,j}. \tag{2.5}$$

Implicit Euler method for Black-Scholes equation.
We discretize equation (1.12) by a fully implicit method. Terms at time point $j+1$ are present on both side of the equation, from which the implicitness of the method.

$$\frac{V_{n,j+1} - V_{n,j}}{\Delta t} \approx \frac{1}{2}\sigma^2 S_n^2 \frac{V_{n+1,j+1} - 2V_{n,j+1} + V_{n-1,j+1}}{\Delta S^2} + rS_n \frac{V_{n+1,j+1} - V_{n-1,j+1}}{2\Delta S} + rV_{n,j+1}. \tag{2.6}$$

Crank-Nicolson method for Black-Scholes equation.
We discretize equation (1.12) by the Crank-Nicolson method, which is an average of the explicit and implicit methods. To approximate the solution, the method will involve six different nodes per time.

$$\begin{aligned}\frac{V_{n,j+1} - V_{n,j}}{\Delta t} \approx &\\ &\frac{1}{4}\sigma^2 S_n^2 \frac{V_{n+1,j+1} - 2V_{n,j+1} + V_{n-1,j+1}}{\Delta S^2} + \frac{1}{4}\sigma^2 S_n^2 \frac{V_{n+1,j} - 2V_{n,j} + V_{n-1,j}}{\Delta S^2} \\ &+ rS_n \frac{V_{n+1,j+1} - V_{n-1,j+1}}{2\Delta S} + rS_n \frac{V_{n+1,j} - V_{n-1,j}}{2\Delta S} + \frac{1}{2} rV_{n,j+1} + \frac{1}{2} rV_{n,j}.\end{aligned}$$

Alternate Direction implicit method for Heston equation.
Alternating direction implicit (ADI) scheme has been originally developed to solve parabolic equation such as the heat diffusion equation. It has the advantage of creating banded matrices with a lower width with respect to other finite difference methods such as Crank-Nicolson scheme. It has been proved that this method is unconditionally stable and second order in time and space [11].

With respect to the Black-Scholes case, Heston assumes a stochastic volatility process: we are approximating the solution over a grid $[Sx\nu]xT$, with equidistanced volatility and underlying points S_j and ν_i, at each timestep t_n. We can apply the ADI scheme to Heston PDE (1.26) and discretize the equation in the following way [27]:

$$\begin{aligned}\frac{V^n_{j,i} - V^{n-1}_{j,i}}{\Delta t} \approx \Bigg[& (S_j)^2 \nu_i \frac{V_{j+1,i} - 2V_{j,i} + V_{j-1,i}}{2\Delta^2 S} \\ & + \rho\sigma S_j \nu_i \frac{V_{j+1,i+1} + V_{j-1,i-1} - V_{j-1,i+1} - V_{j+1,i-1}}{4\Delta S \Delta \nu} \\ & + \sigma^2 \nu_i \frac{V_{j,i+1} - 2V_{j,i} + V_{j,i-1}}{2\Delta^2 \nu} + r S_j \frac{V_{j+1,i} - V_{j-1,i}}{2\Delta S} \\ & + k(\theta - \nu_i) \frac{V_{j,i+1} - V_{j,i-1}}{2\Delta \nu} - r V_{j,i} \Bigg]\end{aligned} \tag{2.7}$$

where $V^n_{i,j}$ is the approximate solution at time t_n computed at nodes (S_j, ν_i). In time direction, we apply the classical *theta scheme* as described in section (2.46).

FDM approximations for Greeks

Following market practice, we choose a *forward difference* approximation for the first derivative and a *central difference* approximation for the second derivative:

$$\Delta := \frac{\partial V(\mathbf{S})}{\partial \mathbf{S}} \approx \frac{V(\mathbf{S} + \Delta\mathbf{S}) - V(\mathbf{S})}{\Delta \mathbf{S}}, \tag{2.8}$$

$$\Gamma := \frac{\partial V^2(\mathbf{S})}{\partial \mathbf{S}^2} \approx \frac{V(\mathbf{S} + \Delta\mathbf{S}) - 2V(\mathbf{S}) + V(\mathbf{S} - \Delta\mathbf{S})}{\Delta \mathbf{S}^2}. \tag{2.9}$$

2.2 Meshfree methods

Meshfree methods have become extremely popular in the last years in several fields, such as Mathematics and Engineering. One of the main reasons to opt for this type of methods is the need for a tool which can handle high-dimensional problems. Furthermore, they are suitable for more complex geometries of the domain considered. Eventually, as the name itself may suggest, they are independent of the generation of a mesh, thus avoiding its computational cost.

Nowadays typical applications of the two most famous meshfree methods, *radial basis functions* and *moving least squares*, can be found in engineering, geodesy, geophysics, meteorology, through the formulation and resolution of a *scattered data problem*. Also, there exist meshfree formulations to solve *partial differential equations* in physics, chemistry, financial engineering. Moreover, these methods are used in nanotechnology, in *non-uniform sampling* for medical imaging and tomography. Applications as neural networks, for data mining and optimization have also been proposed. We can summarize the main historical landmarks as:

- In 1960s, the inverse distance weighted *Shepard* method was developed and published by Donald Shepard [47].
- In 1970s, *multiquadrics* were introduced by Rolland Hardy [26]. Also, *thin plate splines* [25] and *surface splines* [36], which are today known as *polyharmonic splines*, appeared in literature for the first time.
- In 1981, the first paper regarding *moving least squares method* was published by Lancaster and Šalkauskas [32].
- Finally, in the 1990s, the first *compactly supported* radial basis functions were introduced by Shaback and Wendland [44], [56].

2.2.1 Multivariate Scattered Data Interpolation

Historically, the reason to develop *Radial basis functions* (RBFs) methods is the need to solve *Multivariate Scattered Data interpolation* problems: given a set of data with no special structure (*scattered*), we want to find a rule which provides information regarding the studied process also at points different from the ones at which we measure the input data. Formalizing [9], [18]:

Definition 2.2.1. Problem $(\mathcal{P})$: given the data $(\mathbf{x}_j, y_j) \in \mathbb{R}^d \times \mathbb{R}$, $j = 1, \ldots, N$, find a continuous function u_f depending on f such that $u_f(\mathbf{x}_j) = y_j$, $j = 1, \ldots, N$.

If u_f is a linear combinations of functions ϕ_k, i.e.:

$$u_f(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \phi_k(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^d \tag{2.10}$$

the problem $(\mathcal{P})$ can be reduced to solving a linear system of the kind:

$$A\boldsymbol{\alpha} = \mathbf{y} \tag{2.11}$$

with A as *interpolation matrix* such that $A_{jk} = \phi_k(\mathbf{x}_j)$, $j, k = 1, \ldots, N$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]^T$ and $\mathbf{y} = [y_1, \ldots, y_N]^T$.
In general, solving the system (2.47) is a serious problem, depending on the non-singularity of the interpolation matrix A. Nevertheless, one of the main advantages of using radial basis functions is to go around this issue: we will later define special RBFs which have also the important property of being **positive definite**. Such interpolant functions will guarantee the solvability of system (2.47).
Another important property of multivariate scattered data problems is to be highly dependent on the input data. Indeed, the reconstruction of multivariate functions from data is possible only if the space providing the trial functions is not fixed in advance [45], but is data-dependent. This is a consequence of *Haar-Mairhuber-Curtis* theorem. Let us start with the definition of a Haar space:

Definition 2.2.2. Let the finite-dimensional linear function space $\mathbb{V} \subset \mathbb{C}(\Omega)$, with $\Omega \subset \mathbb{R}^d$, have a basis $\{V_1, \ldots, V_N\}$. Then $\mathbb{V}$ is a **Haar space** of dimension N on Ω if

$$det(A) \neq 0$$

for any set of distinct points $\mathbf{x}_i$ in Ω, where A is the matrix such that $A_{i,j} = V_j(\mathbf{x}_i)$.

We can now state the following theorem:

Theorem 2.1. Haar-Mairhuber-Curtis
If $\Omega \subset \mathbb{R}^d$ with $d \geq 2$ contains an interior point, then there exist no Haar spaces of continuous functions except for the 1-dimensional case.

Proof. Let us suppose that $\mathbb{V}$ is a Haar space of dimension $N \geq 2$ on Ω, with a basis $\{V_1, \ldots, V_N\}$ and let $d \geq 2$. Let $\mathbf{x}_i$ be distinct N points in Ω and A the matrix defined by $A_{i,j} = V_j(\mathbf{x}_i)$. Then we have by definition:

$$det(A) \neq 0.$$

Since Ω contains an interior point by assumption, there exists a path that connects the points $\mathbf{x}_1$ and $\mathbf{x}_2$ without passing through the other points of the defined set. Therefore we can exchange the position of the two points continuously along the path, still without moving the other points $\mathbf{x}_{i \neq 1,2}$. As a consequence, the first and second row of the determinant of A can be exchanged and the determinant has changed sign. Being $det(A)$ a continuous function of $\mathbf{x}_1$ and $\mathbf{x}_2$, the determinant must have assumed the value 0, which contradicts our hypothesis.

Examples of data-dependent spaces of multivariate functions are generated by shifted and scaled instances of radial basis functions.

2.2.2 Radial Basis Functions

Definition 2.2.3. We define **radial basis function** any multivariate function with the following property:

$$\Phi(\mathbf{x}) = \phi(\|\mathbf{x}\|_2) = \phi(r), \quad \mathbf{x} \in \mathbb{R}^d, \tag{2.12}$$

where ϕ is a scalar function and $r = \|\mathbf{x}\|_2$ is called radius.

The unknown function u_f in (2.10) can be written as a linear combination of RBF:

$$u_f(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \phi(\|\mathbf{x}\|_2), \qquad \mathbf{x} \in \mathbb{R}^d \tag{2.13}$$

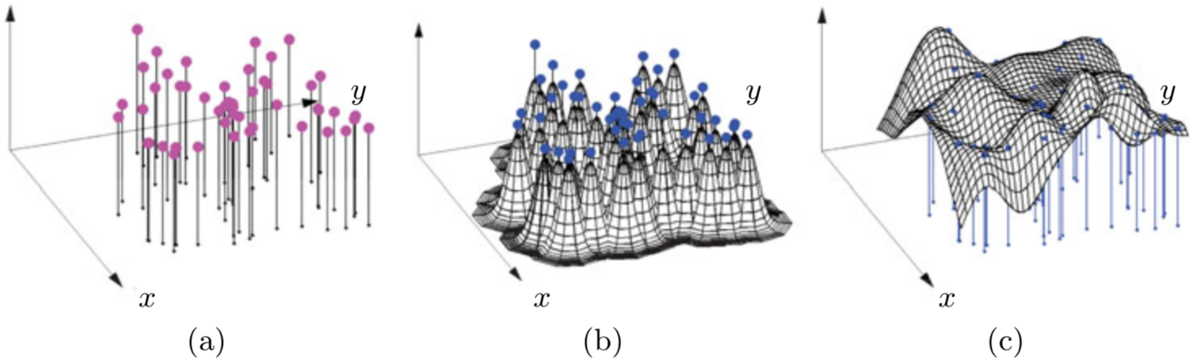


FIGURE 2.1: Example of RBF methods application: in (a) scattered data are shown; in (b) RBF are applied to the knots; in (c) the interpolation produces the final results [20]

or, more generally, as a linear combination of translated RBFs:

$$u_f(\mathbf{x}) = \sum_{k=1}^{N} \alpha_k \phi(\|\mathbf{x} - \mathbf{y}_k\|_2), \qquad \mathbf{x}, \mathbf{y}_k \in \mathbb{R}^d \tag{2.14}$$

where $\mathbf{y}_k$ are called centers. Since there are no assumptions on the position of the centers, RBF related methods are purely meshless.
Given the initial data $(\mathbf{x}_j, u_f(\mathbf{x}_j)) \in \mathbb{R}^d \times \mathbb{R}$, the coefficient matrices

$$A_{\mathbf{x}} := (\phi(\|\mathbf{x}_j - \mathbf{y}_k\|))_{j,k} \tag{2.15}$$

are called **kernel matrices**.

We now define the concept of positive-definiteness, which allows the solvability of the system (2.47). Indeed, if the considered ϕ satisfies this property, the matrix A is non-singular and the system can be solved.

Definition 2.2.4. A radial basis function ϕ on $[0, +\infty)$ is **positive definite** on $\mathbb{R}^d$, if for all choices of sets $X := \{x_1, \ldots, x_m\}$ of finitely many points in $\mathbb{R}^d$ and arbitrary m the symmetric $m \times m$ matrices A_x of (2.15) are positive definite.

Not all radial basis functions are positive-definite. Some of them fail to be and, in this case, polynomials of a certain degree must be added to the approximating function: these RBF will be called **conditionally positive-definite** of order Q, where $Q - 1$ is the degree of the polynomial to add. We provide in table (2.1) a list of the most widely known RBF.

We eventually provide a theorem that summarizes some usefeul properties of positive-definite functions (for the proof, we refer to [17]):

Theorem 2.2. Positive-definite functions properties
Some basic properties of positive definite functions are:

RBF name	RBF	Q
Gaussian	$exp(-r^2)$	0
Inverse Multiquadrics	$(1+r^2)^{\frac{\beta}{2}}, \beta < 0$	0
Matern or Sobolev	$K_\nu(r)r^\nu, \nu > 0$	0
Multiquadrics	$(1)^{\lceil\frac{\beta}{2}\rceil}(1+r^2)^{\frac{\beta}{2}}, \beta > 0, \beta \notin 2N$	$\lceil\frac{\beta}{2}\rceil$
Polyharmonics	$(1)^{\lceil\frac{\beta}{2}\rceil}r^\beta, \beta > 0, \beta \notin 2N$	$\lceil\frac{\beta}{2}\rceil$
Thin-plate spline	$r^2 logr$	2

TABLE 2.1: $Q-1$ is the degree of the polynomial to add to the function to make it positive definite. RBF with $Q=0$ are positive definite [45].

- Non negative finite linear combinations of positive definite functions are positive definite. If $\Phi_1, \ldots, \Phi_n$ are positive definite on $\mathbb{R}^d$ and $c_j \geq 0$, with $j = 1, \ldots, n$, then
$$\Phi(\mathbf{x}) = \sum_{j=1}^{d} c_j \Phi_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \tag{2.16}$$
is also positive definite. Moreover, if at least one of the Φ_j is strictly positive definite definite and the corresponding $c_j > 0$ then Φ is also strictly positive definite.
- $\Phi(\mathbf{0}) \geq 0$.
- $\Phi(-\mathbf{x}) = \overline{\Phi(\mathbf{x})}$.
- Any positive definite function is bounded: $|\Phi(\mathbf{x})| \leq \Phi(\mathbf{0})$.
- If Φ is positive definite with $\Phi(\mathbf{0}) = 0$, then $\Phi \equiv 0$.
- The product of (strictly) positive definite function is (strictly) positive definite.

2.2.3 Collocation approach for PDEs

The first appearance in literature of partial differential equations solving by means of RBFs and collocation methods is due to Ed Kansa [31].
This method can be applied to different types of PDEs, such as non-linear elliptic PDEs and time-dependent parabolic or hyperbolic PDEs [37]. Unfortunately already Kansa pointed out that the so-built interpolant matrix could be not well-defined for some center points. To overcome this issue, a similar collocation approach, based on the Hermite interpolation method, was formulated. The Hermite approach ensures the well-posedness of the collocation matrix, but on the other hand it requires smoother interpolating functions.
We can choose to use linear combinations of RBF plus a polynomial of a certain degree, such that we could obtain a pre-defined order of positive definiteness.
To pose the problem, we will use **collocation methods** following [7] and [45]. Let

the following linear Dirichlet boundary value problem:

$$\begin{aligned} Lu &= f^{\Omega} & \Omega &\subset \mathbb{R}^d \\ u &= f^{\gamma} & \gamma &:= \partial\Omega \end{aligned} \tag{2.17}$$

where L is a linear differential or integral operator. The idea behind collocation methods is to discretize the equations and impose a finite number of pointwise conditions:

$$\begin{aligned} Lu(\mathbf{x}_j^{\Omega}) &= f^{\Omega}(\mathbf{x}_j^{\Omega}) & \mathbf{x}_j^{\Omega} &\in \Omega, 1 \le j \le m^{\Omega} \\ u(\mathbf{x}_j^{\gamma}) &= f^{\gamma}(\mathbf{x}_j^{\gamma}) & \mathbf{x}_j^{\gamma} &\in \gamma, 1 \le j \le m^{\gamma} \end{aligned} \tag{2.18}$$

and let m be the total number $m = m^{\gamma} + m^{\Omega}$ of *test points*.
As a consequence, the exact solution of (2.18) will satisfy also (3.3): we have to fix an m-dimensional space of **trial functions** in which different solutions u can be chosen. The issue with collocation methods is now clear: the solvability of the system in (2.18). However, we can see (2.18) as a generalization of problem $(\mathcal{P})$, so that we could think of applying all the RBF theory reviewed until now.
Following Kansa, we can write the general collocation equations with RBFs:

$$\begin{aligned} \Sigma_{k=1}^{N}\alpha_k \Delta\phi(\|\mathbf{x}_j^{\Omega} - \mathbf{y}_k\|_2) + \Sigma_{l=1}^{Q}\beta_l \Delta p_l(\mathbf{x}_j^{\Omega}) &= f^{\Omega}(\mathbf{x}_j^{\Omega}) & 1 \le j \le m^{\Omega} \\ \Sigma_{k=1}^{N}\alpha_k \phi(\|\mathbf{x}_j^{\gamma} - \mathbf{y}_k\|_2) + \Sigma_{l=1}^{Q}\beta_l p_l(\mathbf{x}_j^{\gamma}) &= f^{\Omega}(\mathbf{x}_j^{\gamma}) & 1 \le j \le m^{\gamma} \\ \Sigma_{l=1}^{Q}\alpha_l p_l(\mathbf{y}_k) &= 0 & 1 \le l \le Q \end{aligned}$$

which form a linear $n \times n = (m^{\Omega} + m^{\gamma} + Q) \times (m^{\Omega} + m^{\gamma} + Q)$ system of equations. The system can be converted to a symmetric one by choosing the same test points $\mathbf{x}_j^{\Omega}$ and $\mathbf{x}_j^{\gamma}$ as trial points, in place of the $\mathbf{y}_k$. We indeed would define:

$$u(\mathbf{x}) := \Sigma_{k=1}^{m_{\Omega}}\alpha_k^{\Omega}\Delta\phi(\|\mathbf{x} - \mathbf{x}_j^{\Omega}\|_2) + \Sigma_{k=1}^{m^{\gamma}}\alpha_k^{\gamma}\phi(\|\mathbf{x} - \mathbf{x}_j^{\gamma}\|_2) + \Sigma_{l=1}^{Q}\beta_l p_l(\mathbf{x}) \tag{2.19}$$

obtaining a symmetric square linear system. The convergence of the method was studied by Franke and Shaback [22]: they found out the solution is required to be very smooth. An important result on the convergence of RBF collocation is due to Wendland [57]:

Theorem 2.3. Error Bound for Elliptic PDE solving via RBF Collocation
Let Ω be a polygonal and open subset of $\mathbb{R}^d$. Let $\mathbb{L}$ be a non-null, second order, linear elliptic differential operator with coefficients in $\mathbb{C}^{2(k-2)}(\bar{\Omega})$ that either vanish on $\bar{\Omega}$ or have no zeros. Let us suppose also that $\Phi \in \mathbb{C}^{2k}(\mathbb{R}^d)$ is a strictly positive

definite function and that the boundary value problem

$$\begin{aligned} Lu &= f^{\Omega} \qquad \Omega \subset \mathbb{R}^d \\ u &= f^{\gamma} \qquad \gamma := \partial\Omega \end{aligned} \tag{2.20}$$

has a unique solution $u \in \mathbb{N}_{\Phi}(\Omega)$ for a given f. Let $\bar{u}$ be the approximate collocation solution. Then:

$$||u - \bar{u}||_{L_\infty(\Omega)} \leq Ch^{k-2}||u||_{\mathbb{N}_{\Phi}(\Omega)}$$

for all sufficiently small h, where $\mathbb{N}_{\Phi}(\Omega)$ is the native space of Φ and h is the largest of the filling distances in the interior of Ω and its boundary.
As a consequence, a good distributions of centers should imply a fill distance smaller on the boundary than in the interior. The space we work with is also an important choice: optimality theorems tell us that, in native spaces of RBFs, the RBFs provide the *best* interpolant of a given data function [17].

2.2.4 Least Squares Radial Basis Functions

A different approach from collocation is using least squares techniques in combination with radial basis functions. This approach has the computational advantage of avoiding the system resolution required by the collocation method.
This method avoids interpolation and is useful in case the data are contaminated by noise. The idea is to use the theoretical results according to which the kernel interpolant minimizes the native space norm. We want to determine a function u such that:

$$u = \sum_{j=1}^{n} c_j \Phi_j(\mathbf{x}, \mathbf{x}_j). \tag{2.21}$$

To find the coefficients c_j we need to minimize the quadratic form:

$$\frac{1}{2}\mathbf{c}^T Q\mathbf{c} \tag{2.22}$$

where Q is a positive definite matrix that satisfies the linear constraint with given $\mathbf{f}$:

$$A\mathbf{c} = \mathbf{f}. \tag{2.23}$$

This quadratic form can be exactly (excluding the the $\frac{1}{2}$ factor) the native space norm of the interpolant function u. Equation (2.22) can be interpreted as a minimization problem by introducing Lagrange multipliers $\boldsymbol{\lambda}$ for the convex functional:

$$\frac{1}{2}\mathbf{c}^T Q\mathbf{c} - \boldsymbol{\lambda}^T[A\mathbf{c} - \mathbf{f}]. \tag{2.24}$$

Consequently we obtain:

$$\begin{aligned} Q\mathbf{c} - A^T\boldsymbol{\lambda} &= 0 \\ A\mathbf{c} - \mathbf{f} &= 0. \end{aligned}$$

Therefore, being Q invertible by hypothesis we can apply Gaussian elimination to find:

$$\begin{aligned} \boldsymbol{\lambda} &= (AQ^{-1}A^T)^{-1}\mathbf{f} \\ \mathbf{c} &= Q^{-1}A^T(AQ^{-1}A^T)^{-1}\mathbf{f}. \end{aligned}$$

The constraint implies that $u(\mathbf{x_i}) = f_i$ and the Lagrange multipliers are nothing else but:

$$\boldsymbol{\lambda} = A^{-1}\mathbf{f} \tag{2.25}$$

with

$$\mathbf{c} = \boldsymbol{\lambda}. \tag{2.26}$$

Eventually, we call the **regularized least squares approximation** problem the following:

$$\frac{1}{2}\mathbf{c}^T Q\mathbf{c} + \omega\sum_{j=1}^{n}(u(\mathbf{x_j}) - f_j) \Leftrightarrow \frac{1}{2}\mathbf{c}^T + \omega(A\mathbf{c} - \mathbf{f})^T(A\mathbf{c} - \mathbf{f}), \tag{2.27}$$

where the paramter ω controls the accuracy in points fitting and the quadratic form controls the accurancy of the fitting function itself.
In RBFs formulation, u would be the interpolant function built as a linear combination of RBFs $\boldsymbol{\Phi}$ and coefficients $\mathbf{c}$, which are the least squares solution of the constraint $A\mathbf{c} = \mathbf{f}$. Therefore, we minimize the quantity:

$$||u - \sum_{j=1}^{n} f^2(\mathbf{x}_j)||_2^2. \tag{2.28}$$

2.2.5 Localization techniques

In case of large systems (2.47), **localization** techniques are necessary. By means of localized basis functions, the system coefficient matrix becomes sparse and easily solvable. This is possible with RBF if we use positive definite scaled functions with compact support.
The most famous compactly supported radial basis function are the **Wendland functions**, introduced by Wendland in 1995 [56]. The first and most known Wendland function is:

$$\psi(r) = \begin{cases} (1 - r^4)(1 + 4r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \tag{2.29}$$

which is strictly positive definite and radial in $\mathbb{R}^d$ when $d \leq 3$.
The following theorem groups the Wendland functions and states that these functions are the polynomial functions with the smallest degree that can be compactly supported on $\mathbb{C}^{2k}$ and strictly positive definite and radial on $\mathbb{R}^d$:

Theorem 2.4. Wendland Functions
The functions $\psi_{s,k}$ are strictly positive definite and radial on $\mathbb{R}^d$ and are of the form:

$$\psi_{s,k}(r) = \begin{cases} p^{s,k}(r) & r \in [0,1] \\ 0 & r > 1 \end{cases}$$

with a univariate polynomial $p_{s,k}$ of degree $\lfloor \frac{1}{2} \rfloor + k + 1$. Furthermore, $\psi_{s,k} \in \mathbb{C}^{2k}(\mathbb{R})$ and are unique up to a constant factor, with a polynomial degree which is minimal for given space dimension d and smoothness $2k$.

By scaling the support of the basis functions appropriately, the interpolant matrix becomes sparse. The trick consists in using the shape parameter ϵ: a large ϵ will imply a smaller radius support $r := \frac{1}{\epsilon}$.
Partition of Unity methods (PUM) are a localization technique that allows for fast computation, by splitting the main problem in smaller ones without affecting accuracy.
The method applied to PDEs solving appeared in literature for the first time in 1996, with Babuška and Melenk paper [1].
The idea is to divide the domain in more subdomains, such as in picture (2.2), and different RBFs are associated to each of them.

Definition 2.2.5. A **partition of unity** subject to an open-cover $\{\Omega_r\}$ of Ω is a collection of smooth, non-negative functions $\{\rho_r\}_{r=1}^{R}$, such that their support is contained in each corresponding $\bar{\Omega}_r$ and

$$\sum_{r=1}^{R} \rho_r(\mathbf{x}) = 1, \qquad \mathbf{x} \in \Omega. \tag{2.30}$$

In practice, a local interpolant u_j is constructed by means of RBFs on each Ω_r; consequently the sum of R interpolants will give the final solution [56]:

$$u(\mathbf{x}) = \sum_{r=1}^{R} u_j(\mathbf{x})\rho_r(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \tag{2.31}$$

In particular, if the local interpolant u_j interpolates at $\mathbf{x}_m$, then also the global interpolant will fit that point:

$$\begin{aligned} u(\mathbf{x}) &= \sum_{r=1}^{R} u_j(\mathbf{x}_m)\rho_r(\mathbf{x}_m) \\ &= \sum_{r=1}^{R} f(\mathbf{x}_m)\rho_r(\mathbf{x}_m) = f(\mathbf{x}_m), \end{aligned}$$

where f is the given function that provides the input data. This result is valid since the point $\mathbf{x}_m$ is contained only in one of the subdomains. It is proved [17] that PUM does not affect the global error estimates.

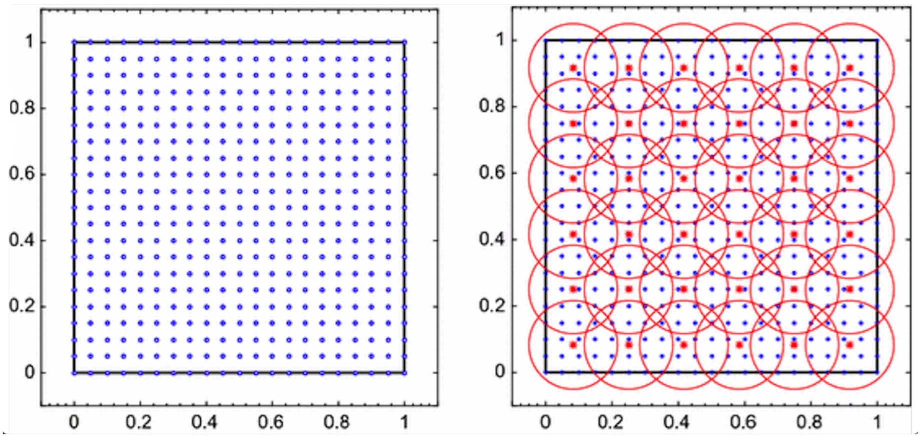


FIGURE 2.2: PUM: example of sub-domains identifications [49].

A common choice in literature is Shepard's method to build the $\{\rho_r\}_{r=1}^{R}$:

$$\rho_r(\mathbf{x}) = \frac{\xi_r(\mathbf{x})}{\sum_{j=1}^{R} \xi_j(\mathbf{x})} \tag{2.32}$$

where the ξ_r are functions with compact support on each Ω_r. Usually, the Wendland function is used as ξ_r:

$$\xi(l) = \begin{cases} (1-l)^4(4l+1) & 0 \le l \le 1 \\ 0 & l > 1 \end{cases} \tag{2.33}$$

The ξ_r will be scaled to match the chosen shape of the subdomain. In our case, we use circular patches so that ξ_r will be applied to the distance from $\mathbf{x}$ to the center of the patch, divided by its radius. Furthermore, we notice that the ξ_r must be at least $\mathbb{C}^k$ on its domain, where k is the order of the derivative required to solve the PDE.

2.2.6 RBF applications to Finance

In this section we finally provide the resolution via RBF for the Black-Scholes and Heston models. We refer to our work in [50], [53], and [52].

Discretization of Black-Scholes equation in time

Any simple method for the discretization in time can be chosen for this purpose. In literature, examples with Crank Nicolson or other FDM schemes are available: we consider the so-called *theta method*, which is a generalization of the already presented FDM. For $\theta = 0$, we obtain the explicit method; for $\theta = 0.5$ we obtain the Crank-Nicolson methods and eventually for $\theta = 1$ we get the fully implicit method.
Let us consider the time interval $[0, T]$ and divide it in N_t steps with length $\delta^n = t^n - t^{n-1}$ for $n = 1, \dots, N_t$. We will make use of a simple backward differential scheme:

$$(I - \alpha_0^n L)V_\star^1 = V_\star^0 \tag{2.34}$$

$$(I - \alpha_0^n L)V_\star^n = \alpha_1^n V_\star^{n-1} - \alpha_2^n V_\star^{n-2} - \alpha_0^n P(V_\star^{n-2}) \tag{2.35}$$

with $V_\star$ solution in the interior of Ω, I the identity operator and with the notation $\theta_n := \frac{\delta^n}{\delta^{n-1}}$:

$$\alpha_0^n := \delta^n \frac{1+\theta_n}{1+2\theta_n}, \qquad \alpha_1^n := \delta^n \frac{(1+\theta_n)^2}{1+2\theta_n}, \qquad \alpha_2^n := \delta^n \frac{\theta_n^2}{1+2\theta_n}. \tag{2.36}$$

The conditions on the boundary are continuously imposed, leading to a the resolution of a linear system at each time step.

Discretization of Black-Scholes equation in space via RBF

We will use RBF methodology to discretize the value of V in the space variable. Let us use the notation $V^n(\mathbf{S})$ to indicate the value of V at time point t_n. Then:

$$\mathbf{v}^n := V^n(\mathbf{S}) = \sum_{k=1}^{n} \alpha_k^n \boldsymbol{\phi}(\epsilon \|\mathbf{S} - \mathbf{S}_k\|_2)$$

$$\mathbf{v}^n = A\boldsymbol{\lambda}^n$$

where ϵ is the *shape parameter*, from which the solution is also dependent. If we consider positive-definite RBF, the matrix A is not singular for distinct knots. Therefore:

$$\boldsymbol{\lambda}^n = A^{-1}\mathbf{v}_n. \tag{2.37}$$

We can finally derive the approximations for the derivatives matrices:

$$\frac{\partial \mathbf{v}^n}{\partial S_k} = A^{(k)}\boldsymbol{\lambda}^n = A^{(k)}A^{-1}\mathbf{v}_n$$

$$\frac{\partial^2 \mathbf{v}^n}{\partial S_k \partial S_i} = A^{(ki)}\boldsymbol{\lambda}^n = A^{(ki)}A^{-1}\mathbf{v}_n.$$

To solve equation (1.7), we now only require the the discretization of both the operator L and the penalty term. Plugging in the formulae we obtained, we have:

$$L\mathbf{v}_n = \left(\frac{1}{2}\sum_{k,i}^{n}\Sigma_{ki}S_kS_iA^{(ki)} + \sum_{k=1}^{n}(r-d_k)S_kA^{(k)} - rA\right)A^{-1}\mathbf{v}_n \tag{2.38}$$

$$P(v_n^{(i)}) = \frac{e\left(rK - \sum_{k=1}^{n} w_k d_k S_k\right)}{v_n^{(i)} + e - q} \tag{2.39}$$

Greeks derivation for Black-Scholes PDE. The space discretization via RBFs allows us to easily calculate the Greeks. Indeed, we can derive the solution $V^n(\mathbf{S})$ with respect to the underlying $\mathbf{S}$.

The Greek **Delta** will be:

$$\frac{\partial V(\mathbf{S})}{\partial \mathbf{S}} = \sum_{i=1}^{n}\boldsymbol{\lambda}^i\frac{\partial\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)}{\partial\mathbf{S}}; \tag{2.40}$$

and the Greek **Gamma** will be:

$$\frac{\partial^2 V(\mathbf{S})}{\partial \mathbf{S}^2} = \sum_{i=1}^{n}\boldsymbol{\lambda}^i\frac{\partial^2\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)}{\partial\mathbf{S}^2}. \tag{2.41}$$

Since the coefficients $\boldsymbol{\lambda}(t)$ are the only time-dependent part, we can also derive the expression for the Greek **Theta**, i.e. the derivative of the solution with respect to time, with negative sign:

$$-\frac{\partial V(\mathbf{S})}{\partial t} = -\sum_{i=1}^{n}\frac{\partial\boldsymbol{\lambda}^i}{\partial t}\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2). \tag{2.42}$$

In finite-difference methods, the Greeks are calculated via difference approximations: the resulting Delta and Gamma are not usually smooth functions. Via RBF formulation, we can calculate the risks by simply multiplying the newly found coefficients $\boldsymbol{\lambda}$ for the corresponding derivative matrix.

If we also apply PU, we need to consider the dependence of the weights on the space variable. Differentiating, we obtain:

$$\frac{\partial V(\mathbf{S})}{\partial \mathbf{S}} = \sum_{i=1}^{n}\sum_{r=1}^{m}\left(\xi_r(\mathbf{S})\boldsymbol{\lambda}^i\frac{\partial\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)}{\partial\mathbf{S}} + \frac{\partial\xi_r(\mathbf{S})}{\partial\mathbf{S}}\boldsymbol{\lambda}^i\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)\right), \tag{2.43}$$

$$\frac{\partial V^2(\mathbf{S})}{\partial \mathbf{S}^2} = \sum_{i=1}^{n}\sum_{r=1}^{m}\boldsymbol{\lambda}^i\left(\xi_r(\mathbf{S})\frac{\partial^2\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)}{\partial\mathbf{S}^2} + 2\frac{\partial\xi_r(\mathbf{S})}{\partial\mathbf{S}}\frac{\partial\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)}{\partial\mathbf{S}}\right. \tag{2.44}$$

$$\left. + \frac{\partial^2\xi_r(\mathbf{S})}{\partial\mathbf{S}^2}\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2)\right), \tag{2.45}$$

$$-\frac{\partial V(\mathbf{S})}{\partial t} = -\sum_{i=1}^{n}\sum_{r=1}^{m}\frac{\partial\boldsymbol{\lambda}^i}{\partial t}\xi_r(\mathbf{S})\boldsymbol{\phi}(\epsilon\|\mathbf{S}-\mathbf{S}_k\|_2). \tag{2.46}$$

Algorithm 1 Greeks computation

a. Calculate and save the derivatives matrices at each step following (2.38);
b. Derive the lambda coefficients by solving the final system in (2.50);
c. Calculate the derivative of lambda coefficients with respect to time;
d. Eventually evaluate the derivatives in Equations (2.43), (2.44), (2.46).

The derivatives matrices with respect to the space variable are already calculated at each step to price the option. As an extra operation, we only need to calculate the derivative of the lambda coefficients with respect to time: this can be done by a simple forward finite-difference method.

Discretization of Heston PDE

We now describe how to solve the Heston PDE (1.26) by means of Radial Basis Functions techniques [18], [31]. We follow and further extend our work published in [50] and [53]. We mainly refer to our recent article [52].
Similarly to the Black-Scholes case, the problem described in equation (1.26) can be reduced to solving a linear system of the kind [17] [31] [49]:

$$A\boldsymbol{\alpha} = \mathbf{y} \tag{2.47}$$

with A as *interpolation matrix* such that $A_{jk} = \phi_k(\mathbf{x}_j)$, $j,k = 1,\dots,N$, $\boldsymbol{\alpha} = [\alpha_1,\dots,\alpha_N]^T$ and $\mathbf{y} = [y_1,\dots,y_N]^T$.

It is clear that we need to discretize the elliptic spatial operator $\mathcal{L}$ and eventually solve the equation in time direction. We fix a point at time t_n and we consider the solution at this particular point, which we will denote by $\mathbf{v}^n$. Therefore, according to RBF theory:

$$\mathbf{v}^n := V(\mathbf{S},\nu,t_n) = \sum_{k=1}^{N} \alpha_k^n \boldsymbol{\phi}(\epsilon\|(\mathbf{x}-\mathbf{x}_{i,j})\|_2) \tag{2.48}$$

$$\mathbf{v}^n = A_{S,\nu}\boldsymbol{\lambda}^n \tag{2.49}$$

where ϵ is the *shape parameter*, from whose choice the solution is also dependent. $\mathbf{x}_{i,j}$ represents the node (S_i,ν_j), which means the RBF radius we are considering is: $r_{i,j} = \sqrt{S_i^2+\nu_j^2}$. If we consider positive-definite RBFs, the matrix $A_{S,\nu}$ is not singular for distinct knots. Therefore we can imply the $\boldsymbol{\lambda}^n$ coefficients by inverting equation (2.49):

$$\boldsymbol{\lambda}^n = A_{S,\nu}^{-1}\mathbf{v}_n. \tag{2.50}$$

Eventually, by assuming that (2.48) is true we can discretize the operator $\mathcal{L}$ as:

$$\mathcal{L} = rS_i \frac{\partial \boldsymbol{\phi}(S_i, v_j)}{\partial S_i} + k(\theta - v_j) \frac{\partial \boldsymbol{\phi}(S_i, v_j)}{\partial v_j} + \frac{1}{2} v_j S_i^2 \frac{\partial^2 \boldsymbol{\phi}(S_i, v_j)}{\partial S_i^2}$$
$$+ \rho \sigma v_j S_i \frac{\partial^2 \boldsymbol{\phi}(S_i, v_j)}{\partial S_i \partial v_j} + \frac{1}{2} \sigma^2 v_j \frac{\partial^2 \boldsymbol{\phi}(S_i, v_j)}{\partial v_j^2} - r \boldsymbol{\phi}(S_i, v_j). \tag{2.51}$$

We only miss the discretization in time, which is done by a classic Crank-Nicolson scheme [58]. Indeed, such a method has been proved to be stable under specific conditions and it is relatively easy to implement [13].

Greeks derivation for Heston PDE. Choosing RBF to price options has the important advantage of providing a fast and accurate method to calculate the Greeks. Indeed, while solving the PDE, we are already calculating the matrices that we will need to directly calculate the derivatives to the underlying $S(t)$, i.e. the Greeks we are interested in.
According to its definitions (2.43) and (2.44), for a particular point in time t_n associated to the coefficients $\boldsymbol{\lambda}^n$, the Greek **Delta** will be:

$$\frac{\partial V}{\partial \mathbf{S}} = \sum_{k=1}^{N} \boldsymbol{\lambda}^n \frac{\partial \boldsymbol{\phi}(\epsilon \| \mathbf{x} - \mathbf{x}_k \|_2)}{\partial \mathbf{S}}; \tag{2.52}$$

and the Greek **Gamma** will be:

$$\frac{\partial^2 V}{\partial \mathbf{S}^2} = \sum_{k=1}^{N} \boldsymbol{\lambda}^n \frac{\partial^2 \boldsymbol{\phi}(\epsilon \| \mathbf{x} - \mathbf{x}_k \|_2)}{\partial \mathbf{S}^2}. \tag{2.53}$$

Clearly, the values of such Greeks also depend on the volatility value v we are considering and we will give as fixed at the time of calculation.

Volatility surface interpolation via RBF

Since RBFs are mainly used as an interpolation method, it is simply natural to apply such method to reconstruct a volatility surface. Given initial market data points, consisting in implied volatilities $\sigma(K, T)$ associated to the corresponding option with Black-Scholes price, we can interpolate said points and imply a volatility value for missing market data.
Formally, given a grid of strikes times maturities of the respective options, i.e. $[KxT]$, with known points $\sigma(K_i, T_j)$ $\forall (K_i, T_j) \in \{K_1, \ldots, K_n\} x \{T_1, \ldots, T_m\}$, where $m, n \in \mathbf{N}$, we aim to find values of $\sigma(K_l, T_h)$ where $l \notin \{1, \ldots, n\}$ and $h \notin \{1, \ldots, m\}$. For sake of simplicity, if we define $\mathbf{x} := (K, T) \in \mathbf{R}^2$ we are required to solve the following

scattered data problem in $\mathbf{R}^2$:

$$\hat{\sigma}(\mathbf{x}) = \sum_{k=1}^{N} c_k \phi(||\mathbf{x} - \mathbf{x}_k||_2) \tag{2.54}$$

where $\sigma(\hat{\mathbf{x}})$ is approximation the volatility function we want to recover; ϕ the chosen radial basis function; $c_k \in \mathbf{R}$ are the coefficients we need to find in order to calculate the linear approximation of the volatility function given above; and $\mathbf{x}_k \in \mathbf{R}^2$ are the chosen centers. Equation (2.54) is a straightforward adaptation of the theory explained in (2.2.1). Hence, in order to obtain the coefficient c_k and compute the volatility function we need to solve the following system:

$$\begin{bmatrix} \phi(||\mathbf{x}_1 - \mathbf{x}_1||_2) & \dots & \phi(||\mathbf{x}_1 - \mathbf{x}_N||_2) \\ \phi(||\mathbf{x}_2 - \mathbf{x}_1||_2) & \dots & \phi(||\mathbf{x}_2 - \mathbf{x}_N||_2) \\ \dots & \dots & \dots \\ \phi(||\mathbf{x}_N - \mathbf{x}_1||_2) & \dots & \phi(||\mathbf{x}_N - \mathbf{x}_N||_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_N \end{bmatrix} = \begin{bmatrix} \sigma(\mathbf{x}_1) \\ \sigma(\mathbf{x}_2) \\ \dots \\ \sigma(\mathbf{x}_N) \end{bmatrix}. \tag{2.55}$$

In order to compute the interpolated matrix, we implement the following algorithm, which can be generalized for a generic two-dimensional RBF interpolation:

Algorithm 2 RBF interpolation for a 2D implied volatility surface

a. Define the chosen RBF in terms of the distance $r \in \mathbf{R}$ between data points;
b. Compute the distance matrix *DM* between the given data points and centers;
c. Compute the distance matrix *DME* between the new evaluation points and centers;
d. Compute the interpolation matrix *I* by applying the chosen RBF to the distance matrix DM;
e. Compute the evaluation matrix *IE* by applying the chosen RBF to the distance matrix DME;
f. Compute the interpolation system solution *ISS* by dividing the interpolation matrix *IM* by the array filled with the volatility input values;
g. Compute the interpolated solution given by the product of the evaluation matrix *IE* and the interpolation system solution *ISS*.

Future work may consist in exploring how RBFs behave in case of extrapolated points and deriving conditions in order to get an arbitrage-free volatility surface: we are currently dealing with such questions in our working paper [54]. Deriving an arbitrage-free volatility surface is still an open question: constraints have been formulated and listed in [28], [41], [19]. Furthermore, this technique could be improved by using a least square approach we mentioned in section (2.22), partly following [24].

2.3 Monte Carlo methods

In a complete and arbitrage-free market, the price of a derivative may be viewed as the discounted expected value of its future payoff. Thus, the valuation of the price translates to the computation of an integral or, more attractively, to the Monte Carlo evaluation of such expectation. Typically, a certain number of paths is simulated for each stochastic parameter involved in the model: in case of high dimensions - i.e. more parameters - the square-root convergence rate of Monte Carlo becomes appealing.

Monte Carlo (MC) is an alternative method to price derivatives which avoids the direct resolution of the PDE associated to the price process. Furthermore, the algorithm is relatively easy to understand and implement especially in case of *path-dependent derivatives*, such as American or Barrier options. Path-dependent derivatives are indeed defined as those derivatives whose value cannot be fully expressed as the expectation of the discounted future value, and depends also on extra features such as exercises strategies or barriers.

The main idea behind Monte Carlo is to calculate the volume of a given set and translate this information into a probability. We can imagine to draw a square of one unit length, containing a circle of the same diameter: by sampling a certain number of points in the square and calculating the number of points which fall within the circle, we are able to estimate the circle's area. The result will be better as the number of sampled points increase, according to the law of large numbers. The relative error may be estimated by the central limit theorem.

More formally, let us consider a real function $h : x \in \Re \longmapsto h(x) \in \Re$, square-integrable over $[0,1] \subset \Re$. We want to calculate the following integral via Monte Carlo:

$$s = \int_0^1 h(x)dx. \tag{2.56}$$

Given a random generator that produces a number n of independent and uniformly distributed random draws u_i from the interval $[0,1]$, the Monte Carlo result for (2.56) is:

$$\hat{s}_n := \frac{1}{n}\sum_{i=1}^{n} h(u_i), \tag{2.57}$$

and according to the strong law of large numbers:

$$\hat{s}_n \longrightarrow s \qquad \text{for} \qquad n \longrightarrow +\infty. \tag{2.58}$$

The error associated with Monte Carlo is clearly:

$$\epsilon_n = \hat{s}_n - s, \tag{2.59}$$

and is proven to be [23] normally distributed, with mean 0 and a standard deviation of $\frac{\sigma}{\sqrt{n}}$, where:

$$\sigma^2 := \int_0^1 (h(x) - s)^2 dx. \tag{2.60}$$

Since we do not know σ, we may value its unbiased, empirical estimator:

$$\hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (h(u_i) - \hat{s}_n)^2}. \tag{2.61}$$

The Monte Carlo error is thus dependent on the number n of draws, following a square-root convergence rate.

2.3.1 Monte Carlo applications to Finance

For this section we mostly refer to our work in [51]. The option pricing can be interpreted as a classical example of *inverse problem*. From a formal point of view, we have two stochastic processes Y_t and X_t, where Y_t represents a phenomenon and X_t its stochastic casual factors, and we suppose that a functional dependence between them, indicated with F_t, exists:

$$Y_t = F_t(X_t, \theta_t),$$

where θ_t is a time-depending vector. An inverse problem aims at the estimation of F_t and θ_t, given a set of observations of Y_t.
In our context, the stochastic process Y_t is identified as the option price, the stochastic process as the underlying, the parameter vector is the risk-free interest rate and the volatility of the underlying. In [51] we aim to determine the function F_t and we assume that in our market the interest rate and the volatility are known constants.
In the following we fix a time interval $[0; T]$, a Brownian motion W_t and a probability space $(\Omega; \mathbb{F}_t; \mathbb{P})$ for all stochastic processes; we denote the constant risk-free interest rate with r. The lognormal risk asset S_t with constant volatility σ and drift μ solves the Geometric Brownian motion stochastic differential equation:

$$dS(t) = \mu S(t)dt + \sigma S(t) dW_t. \tag{2.62}$$

The drift parameter is the difference between the interest rate r and a constant dividend q. We recall the no arbitrage vanilla option price P_0^{bs} at the instant 0 is given by the Black-Scholes formula [4], which we recall for sake of simplicity:

$$P_0^{bs} = \begin{cases} S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2) & \text{for a Call} \\ Ke^{-rT}\Phi(-d_2) - S_0\Phi(-d_1) & \text{for a Put} \end{cases} \tag{2.63}$$

where:

$$d_1 = \frac{\log\left(\frac{S_0}{K}\right)+\left(r+\frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T} \quad \Phi(d) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{d} e^{-\frac{x^2}{2}}\,dx.$$

Let us use the following notation for the payoff function H:

$$H(S_t) = \begin{cases} \max\{S_T - K, 0\} & \text{for a Call} \\ \max\{K - S_T, 0\} & \text{for a Put} \end{cases}$$

In the following we indicate the expected value of a random variable X with $E[X]$ and the indicator function of an interval $]a;b[$ with $1_{]L;U[}(x)$.
The prices of a knock-in and knock-out barrier option at time $t = 0$, with underlying S_t and pay-off function $H(S_t)$, are expressed in the following way:

$$\begin{cases} P_0^{out} = e^{-rT}E\left(H(S_T)\prod_{t\in[0;T]} 1_{]L;U[}(S_t)\right) \\ P_0^{in} = P_0^{bs} - P_0^{out}. \end{cases} \tag{2.64}$$

We decompose the time interval $[0;T]$ with N equidistant points $0 = t_0 < ... < t_{N-1} = T$ and we indicate the value of the underlying at the instant t_n and the discretization step respectively with the symbols S_n and h; for each n, the variables $H(S_N)$ and $1_{]L;U[}(S_n)$ are uncorrelated. Under such assumptions, the following approximations of the knock-out and knock-in barrier option prices are valid:

$$\begin{cases} P_0^{out} \approx e^{-rT}E[H(S_{N-1})]\prod_{n=0}^{N-1} E\left[1_{]L;U[}(S_n)\right] \\ P_0^{in} \approx P_0^{bs} - P_0^{out}. \end{cases} \tag{2.65}$$

At each step n, our numerical procedure consists in approximating the value S_n of the previous system by the mean of M particles and in estimating the price values in (2.65) by a Monte Carlo approach. More in detail, our procedure consists of the following steps:

a. generation of a number M of the particles $S_n^{(m)}$, with $1 \le m \le M$ and $0 \le n \le N-1$;
b. rejection of the underlying values (the particles) with the lowest probability to stay in the interval $]L, U[$;
c. determination of the values in (2.65).

We set $X_t := logS_t$ and we transform the system (2.62), obtaining:

$$\frac{dX_t}{dt} = \mu + \sigma\frac{dW_t}{dt}. \tag{2.66}$$

The numerical solutions $X_n^{(m)}$ of this equation, which represent the logarithms of the particles $S_n^{(m)}$, are found by applying a first-order numerical method as follows:

$$X_{n+1}^{(m)} = \mu_n^m + \sqrt{h}\sigma z_{n+1}^{(m)} \quad z_{n+1}^{(m)} \sim N(0;1) \quad \mu_n^m = X_n^{(m)} + hr$$

Next step consists in (re)sampling such particles. This is achieved by defining two suitable functions g and G in the following way:

$$\begin{aligned} g(X_{n+1}^{(m)}; X_n^{(m)}) &:= \mathbb{P}(X_{n+1}^{(m)} \in]\ln L; \ln U[\mid X_n^{(m)}) \\ &= \frac{1}{\sigma\sqrt{2h\pi}} \int_{\log L}^{\log U} e^{-\frac{(x-\mu_n^m)^2}{2h^2\sigma^2}}\, dx. \end{aligned} \tag{2.67}$$

$$G(X_{n+1}^{(m)}, X_n^{(m)}) \quad := \quad \begin{cases} g(X_{n+1}^{(m)}; X_n^{(m)}), & \text{if} \quad X_{n+1}^{(m)} \in]\log L; \log U[\\ 0, & \text{otherwise.} \end{cases} \tag{2.68}$$

We briefly explain the re-sampling technique. We only choose the particles with the highest values of the function G. At each step, we sum the values of the function G until its sum does not become greater than a drawn uniform number u_k, with k an integer, and we indicate the number of the addends with s_k: at this point, we set all the previous particles equal to s_k-th particle. For each time step, this procedure stops when the sum of such indices s_k is equal to M, in order to consistently obtain a number M of particles.
Eventually, estimators for the knock-out and knock-in price P_{out} and P_{in} have been determined according to the law of large numbers:

$$\begin{cases} P_0^{out} = e^{-rT} \left(\frac{1}{M} \sum_{m=1}^{M} H\left(S_{N-1}^{(m)}\right)\right) \prod_{n=0}^{N-1} \frac{1}{M} \sum_{m=1}^{M} 1_{]L;U[}\left(S_n^{(m)}\right) \\ P_0^{in} = P_0^{bs} - P_0^{out}. \end{cases} \tag{2.69}$$

The input variables for the pricing algorithm are the upper bound T of the time interval, the number N of time steps, the number M of particles at every time step, the barriers L and U, the strike price K, the risk-free interest rate r, the assumed constant volatility σ, the spot value of the underlying S_0, the theoretical Black-Scholes vanilla price P_0^{bs}.
The algorithm will eventually output the knock-out barrier option price P_0^{out} and the knock-in barrier option price P_0^{in}.

Algorithm 3 Algorithm for Barrier Option pricing via Particle Filtering

Require: $T, M, N, L, U, K, r, \sigma, S_0, P_0^{bs}$
Ensure: $P_0^{(out)}, P_0^{(in)}$

$$X_1^{(m)} = \log S_0^{(m)}; \quad h = \frac{T}{N} \quad m = 1, ..., M. \quad \textit{Initialization}$$

for $n = 1, N-1$ **do**
  for $m = 1, M$ **do**

$$\begin{aligned} X_{n+1}^{(m)} &= X_n^{(m)} + hr + \sqrt{h}\sigma z_{n+1}^{(m)}; \quad z_{n+1}^{(m)} \sim N(0;1). \\ \mu_n^m &= X_n^{(m)} + hr. \quad \textit{Generation of the values} \quad X_n^{(m)} \\ g(X_{n+1}^{(m)}; X_n^{(m)}) &= \frac{1}{\sigma\sqrt{2h\pi}} \int_{\ln L}^{\ln U} e^{-\frac{(x-\mu_n^m)^2}{2(h\sigma)^2}} \, dx. \end{aligned}$$

if $X_{n+1}^{(m)} \in]\ln L; \ln U[$ **then**

$$G^1(X_{n+1}^{(m)}; X_n^{(m)}) = g(X_{n+1}^{(m)}; X_n^{(m)}) \quad G(X_{n+1}^{(m)}; X_n^{(m)}) = \frac{G^1(X_{n+1}^{(m)})}{\sum_{k=1}^M G^1(X_{n+1}^{(k)}; X_n^{(k)})}.$$

else $G(X_{n+1}^{(m)}; X_n^{(m)}) = 0$. *Computing of the functions g and G*
end if
end for

for $m = 1, M$ **do** $(G = 0 \quad j = 1 \quad u^{(m)} \sim Uniform(0,1))$
while $G < u^{(m)}$ and $j \leq M$ **do** *Re-sampling of the particles.*

$$G = \sum_{k=1}^{j} G(X_{n+1}^{(k)}); X_n^{(k)}) \quad j \leftarrow j + 1.$$

end while

$$X_{n+1}^{(k)} = X_{n+1}^{(j)}; \quad G(X_{n+1}^{(k)}; X_n^{(k)}) = G(X_{n+1}^{(j)}; X_n^{(j)}) \quad k = 1, ..., j-1.$$

end for
end for

$$\begin{aligned} S_n^{(m)} &= exp(X_n^{(m)}); \quad P_0^{out} = e^{-rT} \left(\frac{1}{M} \sum_{m=1}^{M} H\left(S_{N-1}^{(m)}\right) \right) \prod_{n=0}^{N-1} \frac{1}{M} \sum_{m=1}^{M} 1_{]L;U[} \left(S_n^{(m)}\right) \\ P_0^{in} &= P_0^{bs} - P_0^{out}. \quad \textit{Option pricing} \end{aligned}$$

Chapter 3

Option pricing and Greeks evaluation: numerical evidence

In this chapter we are going to illustrate the main results we have obtained by pricing several types of financial derivatives by means of the described numerical methods, and we will furthermore compare such results.
In section (3.1) we will present the results concerning the classical Black-Scholes framework: we will discuss the error resulting from each technique; we will enumerate and comment the results for pricing and eventually for the Greeks calculation. In section (3.2) we will show the main results regarding the Heston model.

3.1 Numerical results for Black-Scholes model

We provide several numerical experiments: a vanilla American option, an American basket option with two underlyings and eventually a barrier option. We calculated the prices by means of RBF and we studied the change in the result with respect to different shape parameters ϵ. In all experiments we use multiquadrics and Gaussian radial basis functions. We used market data as inputs.
Furthermore, we compare Greeks calculated via finite difference methods and RBF-PUM expressions. In order to do this, we choose a step of circa 0.1, which allows for an accurate FDM result and for a direct comparison between the two methods. Eventually, we present the results of the Particle Filtering algorithm to price a barrier option and we compare them to a standard Monte Carlo algorithm.

3.1.1 Pricing comparison via RBF and FDM

We start by applying the described methods to price an American vanilla option. In figure (3.1) we show the calculated price of an American call option with RBF methods, varying with respect to underlying price. We used real data to price an option with strike price 225 USD on the Apple stock, with maturity 16-11-2018 and pricing time 10-09-2018. We considered a fixed dividend of 2.72%, a fixed market implied volatility of 26.33 and we considered OIS as risk free rate, equal to 198 bps on the valuation date.
We chose a penalty term of 0.05: numerical experiments showed that the shape of

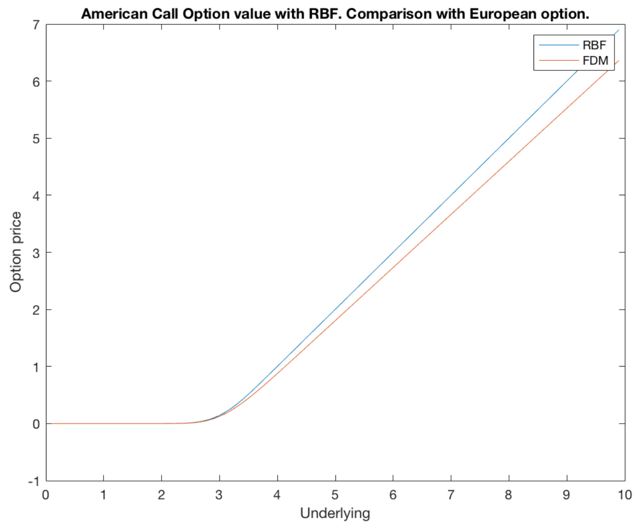


FIGURE 3.1: American Call Option price in USD via RBF methods, compared with Black Scholes classical solution.

the solution becomes sensitive to the penalty term choice if it is bigger than 1. Indeed, in this case the price of the option becomes negative for low underlying prices. Eventually, we chose a shape parameter ϵ equal to 1.5 for Multiquadrics and 0.7 for Gaussians: we noticed that for very low ϵ values the matrix of coefficients A becomes singular. Both basis functions performed well: the choice of applying PU method improves the results in case of Multiquadrics, which otherwise could prove quite unstable [8], [38].
We also added a comparison with the MATLAB function that calculates the value of an European option with the same parameters: opting for a penalty term equal to 0, we recover the European case, in line with theory, as showed in figure (3.2).

We priced an Americal Call Basket Option, with two underlyings having a correlation of $0.94 \approx 1$. We considered as underlyings the stocks of CitiGroup and Morgan Stanley, with implied volatilities of respectively 23.21 and 22.60; respective dividends of 1.41% and 1.05%; a strike price of 44.62 USD, which corresponds to 100% of Morgan Stanley starting price. The risk free parameter is set to 218, which is the value of OIS on the pricing date. We chose a penalty parameter of 10^{-4} and a shape parameter $\epsilon = 1$. Resulting price surface is represented in figure (3.3). In figure (3.4) we show how a bad choice of shape parameters and penalty parameters may easily result in the failure of RBFs interpolation.
Moreover, we price knock-out call barrier options with the same input data of the American call, with one underlying and barriers equal to 600 USD and 700 USD. We show the results in figure (3.5) and (3.6). In the first figure, we can notice the payoff at time $t = 0$ with its typical discontinuity. In the second figure, we see how the discontinuity is *smoothed* way down to the option value at time $t = T$ of expiry.

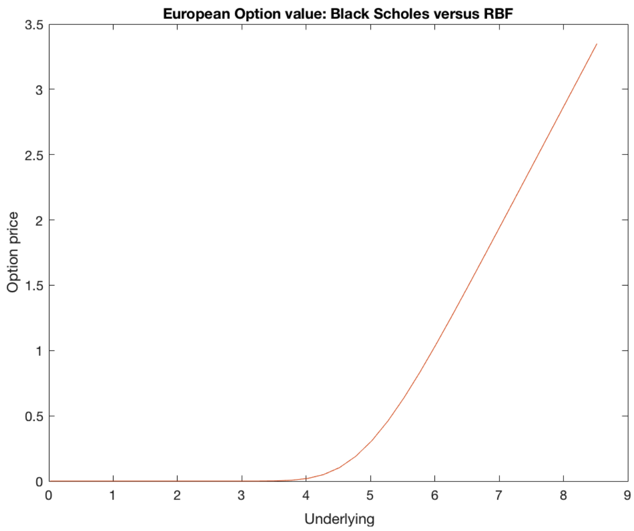


FIGURE 3.2: Comparison between European option with analytical solution of Black Scholes and RBF method with a null penalty term. Price in USD.

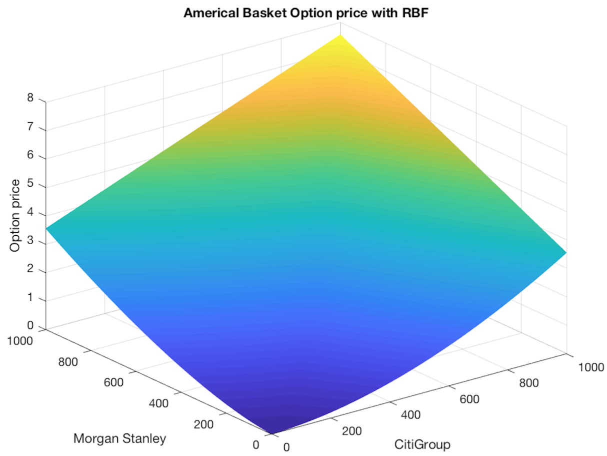


FIGURE 3.3: American call basket option price in USD via RBF.

3.1.2 Pricing comparison via standard MC and sequential MC

In this section we compare the standard Monte Carlo method and the improved sequential Monte Carlo. More in detail, we have studied the pricing problem of a knock in down put with a *single barrier* B (i.e. in our model we have set $L = -\infty$ and $U = B$), with Volkswagen company stocks as underlying and 6M EURIBOR as the risk-free interest rate. We have applied our procedure and compared the values and the ones of a standard Monte Carlo method (as presented in [42]) with the real price $P = 0.36$ € for different values of N and M between 10 and 100. In the following we list all the values of the input variables:

$$(r, q, S_0, B, K, \sigma, T, P_0^{bs}) = (0.0056, 0.00005, 75, 100, 0.09, 1 \text{ year}, 78.46\,€).$$

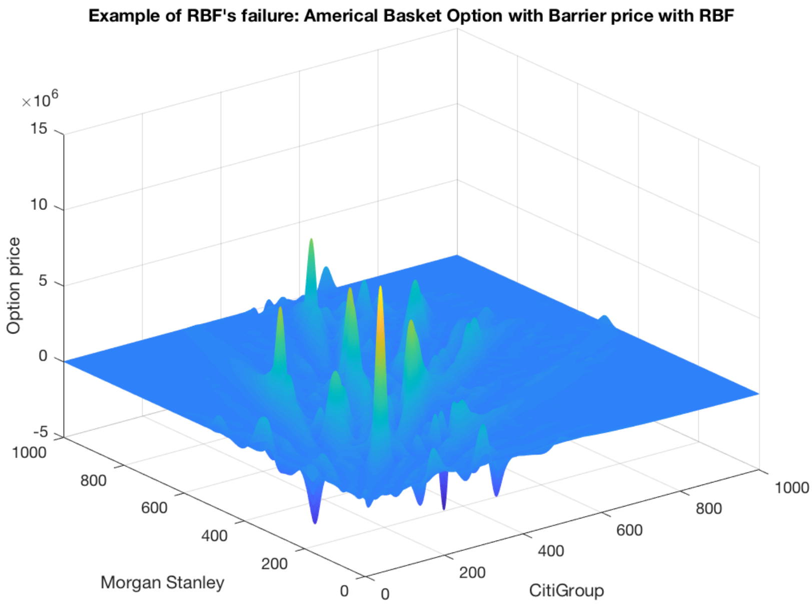


FIGURE 3.4: American call basket option price in USD via RBF: results in case of unsuitable shape and penalty parameters.

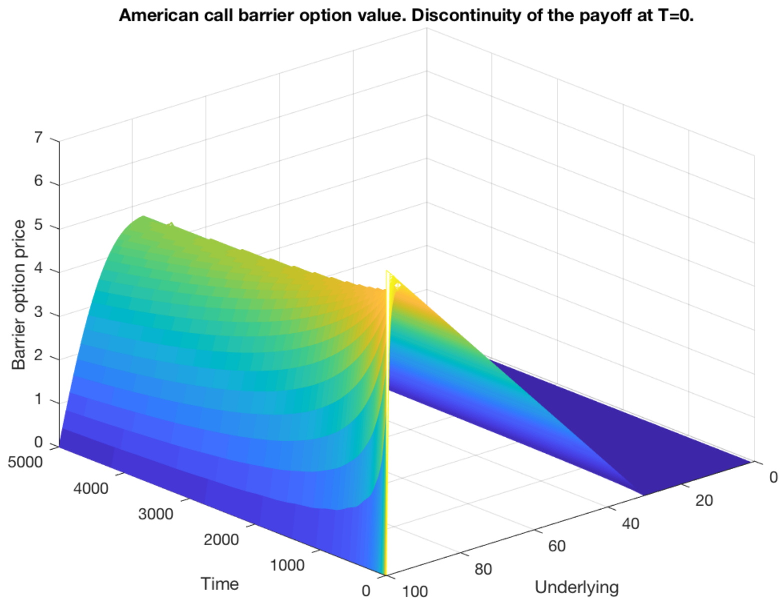


FIGURE 3.5: American call barrier option price in USD via RBF: payoff at time $t = 0$.

We show the results through table (??), constructed in the following way: i) the first and the second column contain respectively the values of N and M; ii) the third, the fourth and the fifth columns contain respectively the estimations of the price by the algorithm described in section (2.3.1) (indicated with *SMC*), the corresponding absolute error values (indicated with *Error SMC*) and the average time to complete

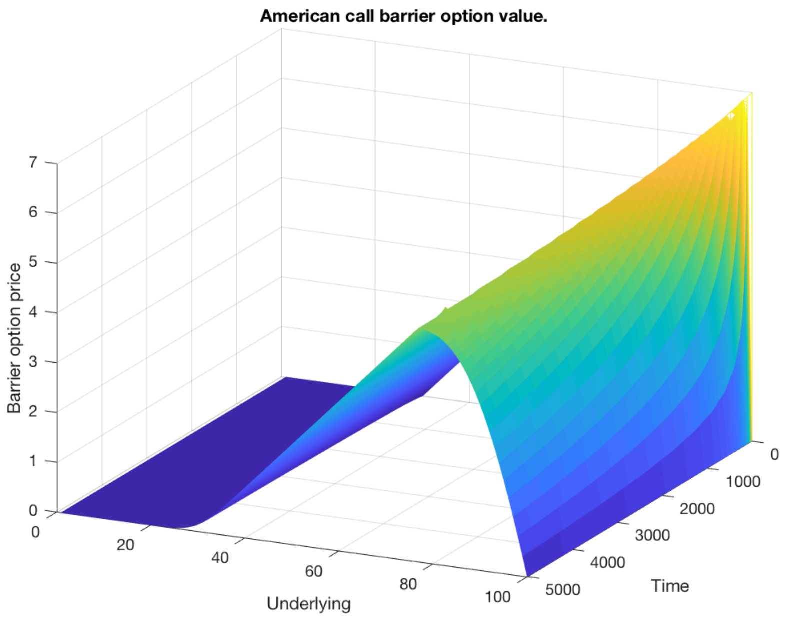


FIGURE 3.6: American call barrier option price in USD via RBF: solution at time of expiry.

a simulation (indicated with *Time SMC*); iii) the sixth, the seventh and the eighth columns contain respectively the approximations of the price by the standard Monte Carlo (indicated with *MC*), the corresponding absolute error values (indicated with *Error MC*) and the average time to complete a simulation (indicated with *Time MC*). All the numerical values have been expressed in Euro.

N	M	SMC	Error SMC	Time SMC (seconds)	MC	Error MC	Time MC (seconds)
100	20	0.23	0.14	0.05	0	0.36	0.04
80	80	0.55	0.19	0.18	0.73	0.37	0.01
80	20	0.35	0.01	0.06	0.51	0.15	0.003
60	50	0.16	0.26	0.09	0.77	0.41	0.004
60	40	0.34	0.02	0.12	0.21	0.15	0.02
60	30	0.37	0.01	0.06	1.43	1.07	0.003
40	50	0.04	0.33	0.07	1.23	0.87	0.003
40	20	0.74	0.39	0.03	1.05	0.7	0.003
20	50	0.43	0.07	0.07	0.34	0.02	0.01
20	20	0.84	0.48	0.03	0.85	0.49	0.003
10	20	0.78	0.42	0.01	1.49	1.13	0.003
10	100	0.38	0.02	0.04	0.83	0.47	0.003
100	100	0.60	0.24	0.43	0.69	0.33	0.01
100	150	0.42	0.07	0.47	1.06	0.70	0.03
150	150	0.60	0.24	0.64	0.33	0.03	0.13
200	150	0.44	0.08	0.83	0.79	0.43	0.02
200	200	0.68	0.31	1.11	0.99	0.63	0.02

TABLE 3.1: Simulations of barrier option prices for different values of N and M.

Our results suggest that:

- for high values of N and M both the sequential and the standard Monte Carlo methods have similar results on average;
- for low values of N and M the sequential Monte Carlo method gives better results than a standard Monte Carlo procedure.

In conclusion, our method is more suitable than a standard Monte Carlo with respect to low values of N and M: this is due in particular to three factors: the discretization of the underlying, the re-sampling technique and the chosen price estimator.

3.1.3 Greeks calculation

In this section we show the main results concerning Greeks calculation. We calculated Delta, Gamma and Theta for different kind of options and compared them with the corresponding finite-difference results.
We remind that in case of Monte Carlo pricing, Greeks are usually calculated via finite-difference method as well: thus, we will focus on Greeks generated by means of RBF and FDM.
Especially for cases in which finite-difference is not sufficient to provide a smooth solution, RBFs show themselves to be definitely superior.
In figures (3.7, 3.8, 3.9) we show Delta, Gamma and Theta functions calculated by means of RBF. The solutions are smooth: especially in case of Gamma, we can notice

an improvement of the smoothness around the peak of the function, when compared with the finite-difference solution.

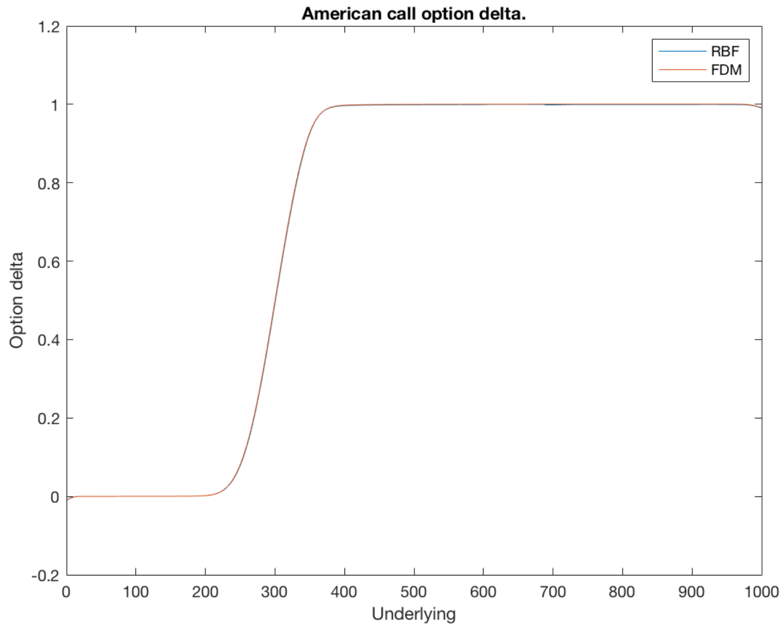


FIGURE 3.7: American call option delta calculated via RBF and FDM.

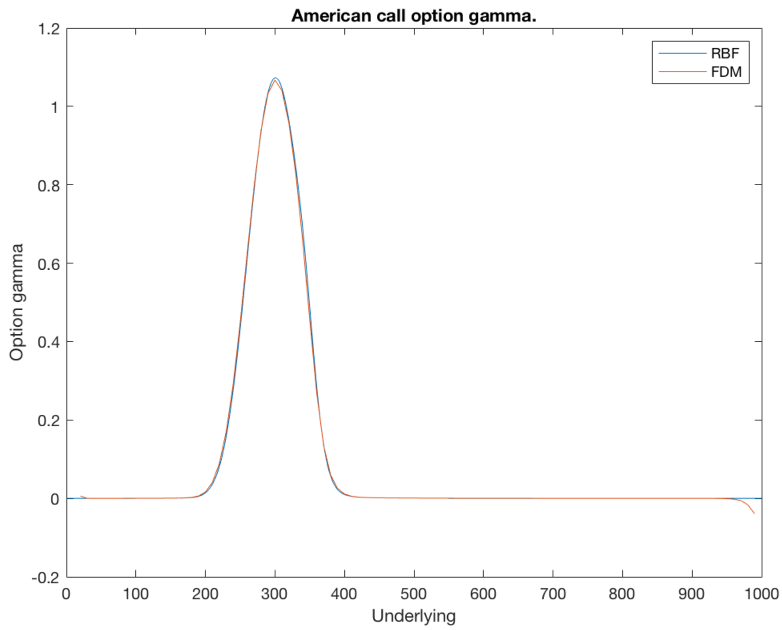


FIGURE 3.8: American call option gamma calculated via RBF and FDM.

Eventually, we show our results in case of a knock-out barrier option. It is a well-known issue to provide smooth Greeks for this kind of product because of the discontinuity caused by the barrier. Via finite-difference methods, we hardly get a

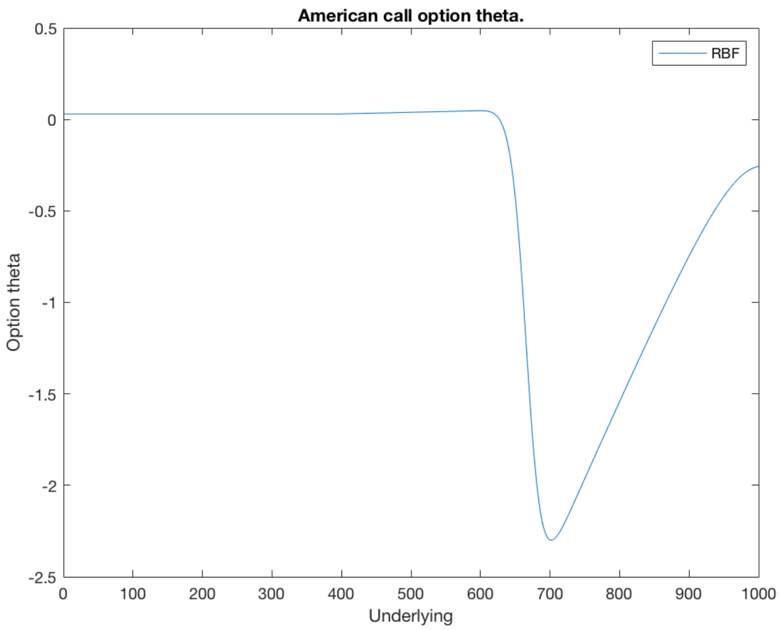


FIGURE 3.9: American call option theta calculated via RBF.

continuous function; by means of RBF formulae, we indeed obtain satisfying results (3.10, 3.11). We also show in Figure (3.12, 3.13) the finite-difference solutions for Delta and Gamma.

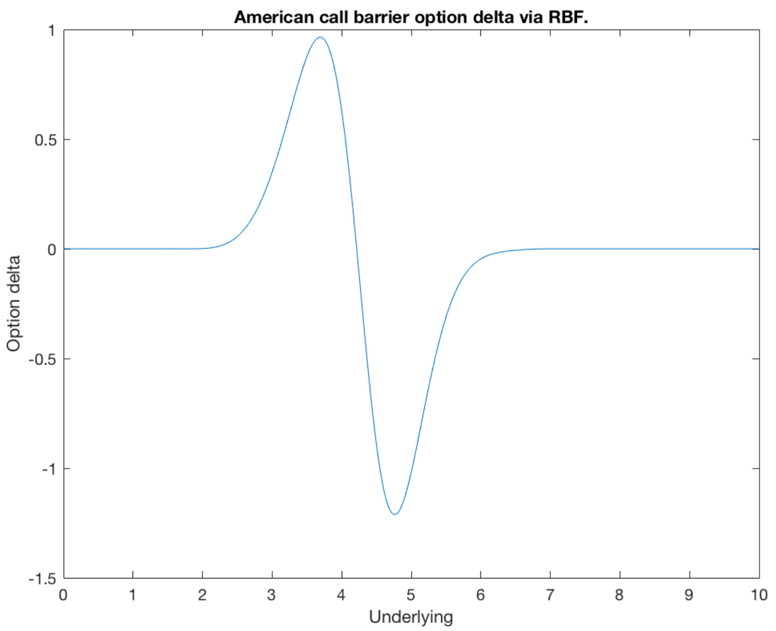


FIGURE 3.10: American call knock-out barrier option delta calculated via RBF. Barrier level at 600 USD.

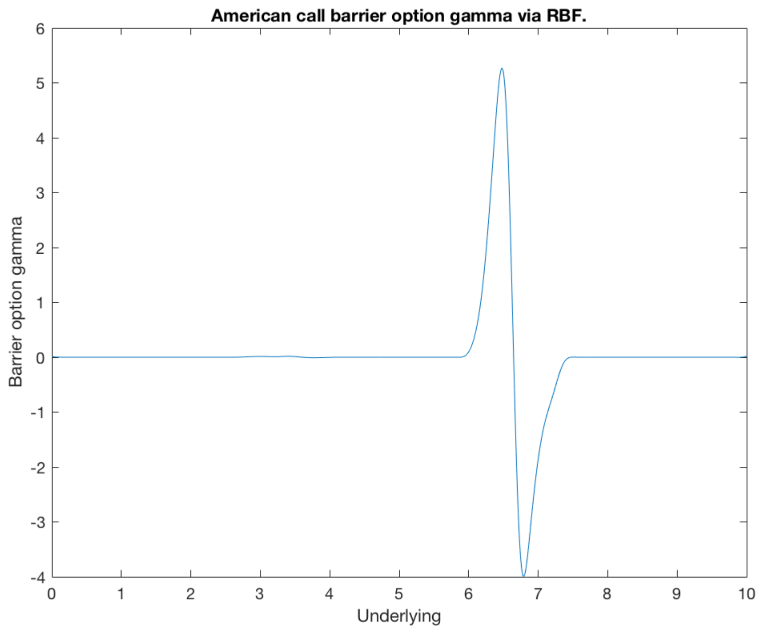


FIGURE 3.11: American call knock-out barrier option gamma calculated via RBF. Barrier level at 700 USD.

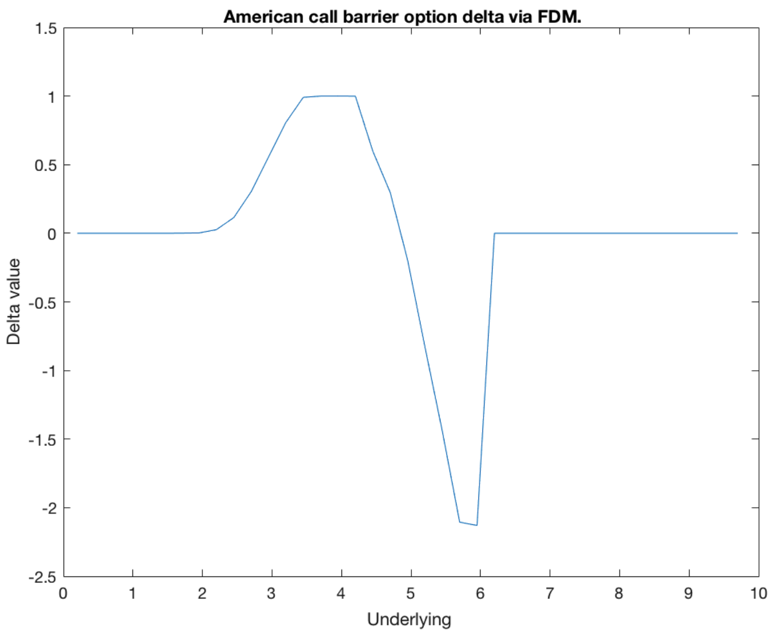


FIGURE 3.12: American call knock-out barrier option delta calculated via FDM. Barrier level at 600 USD.

3.1.4 Error estimation

Let us denote the time step as Δt. For finite difference methods, the error can be estimated as:

$$E^{FD}(t) = O(\Delta t) + O(\Delta \mathbf{S}^2), \tag{3.1}$$

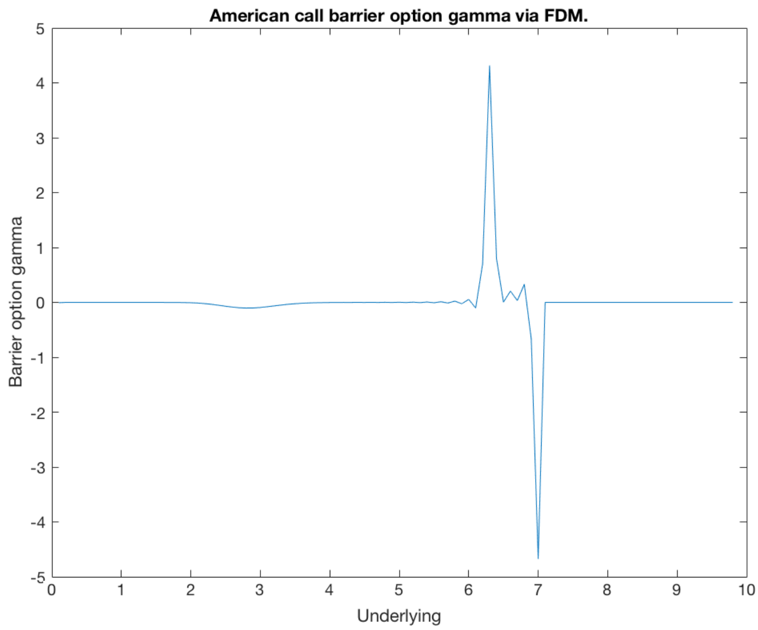


FIGURE 3.13: American call knock-out barrier option gamma calculated via FDM. Barrier level at 700 USD.

for forward difference methods, while for central difference methods we have:

$$E^{CD}(t) = O(\Delta t^2) + O(\Delta \mathbf{S}^2). \tag{3.2}$$

In [43], the error of RBF-PU methods with a fixed partition number is estimated as:

$$||E^{RBF-PU}(t)||_\infty \leq Ce^{-\frac{\gamma}{\sqrt{\Delta S}}} \max_{0\leq\tau\leq t} \max_i ||V(\tau)||_{\mathcal{N}}, \tag{3.3}$$

where γ is a constant which is taken as the minimun over the patches. Equation (3.3) results in an exponential convergence rate for RBF-PU methods, in optimal conditions. Thus, for the same space step $\Delta\mathbf{S}$ and time step Δt, we expect a better result for RBF-PU than for finite-differences methods.

It should be pointed out that FDM may be applied to solve a PDE derived from (1.12), whose solution is already one of the defined Greeks. Therefore, FDM would imply a better solution than the proposed one in terms of error and smoothness. However, it is computationally too expensive to solve a PDE for each Greek: this is the reason why we compare RBF with the proposed approach, which is in practice the most common choice.

3.1.5 Implied Volatility Interpolation via RBF

We conclude the section regarding the Black-Scholes model results with the BS-implied volatility surface reconstruction via RBF methods.

By selecting several, common basis functions such as Gaussians and inverse multiquadrics, we were not able to perfectly recover the surface by also matching the values of the initial volatility points. By trial and error - and also helped by the idea of how the surface behaves at the limit regions - we finally obtained a smooth surface via thin plate splines or via multiquadrics: such functions better fit the final type of shape we aim for. We chose a shape parameter quite high: $\epsilon = 40$, while in our pricing experiments we dealt with very small epsilon values.
We used Apple stocks data, and considered the BS-implied volatility for strikes varying between 80% and 120% of the stock price on valuation date as shown in table (3.2) and in figure (3.14). Resulting interpolated surface is shown in figure (3.15). Results seem to be promising given the sufficiently monotone and convex surface recovered via RBF. Further improvements would involve data filtering in order to filter out any initial data point which already violates theoretical constraints for an arbitrage-free volatility surface.

Maturity/Strike	**80** %	**90** %	**95** %	**97.5** %	**100** % **(ATM)**	**102.5** %	**105** %	**110** %	**120** %
05-Feb-21	75.44	56.98	51.06	50.45	50.62	51	51.79	56.78	69.21
12-Feb-21	62.77	50.81	47.46	47.28	47.34	47.33	47.43	49.57	57.58
19-Feb-21	56.89	47.27	45.16	44.85	44.64	44.5	44.57	45.92	51.45
26-Feb-21	54.15	46.74	44.88	44.61	44.53	44.52	44.58	45.29	49.87
05-Mar-21	52.34	46.39	44.85	44.55	44.44	44.45	44.53	45	48.33
12-Mar-21	51.71	45.80	44.77	44.49	44.3	44.17	44.12	44.4	47.1
19-Mar-21	50.45	45.49	44.71	44.48	44.3	44.17	44.11	44.29	46.33
16-Apr-21	46.82	43.82	43.29	43.15	43.05	42.99	42.97	43.09	44.19
18-Jun-21	45.09	43.20	42.88	42.78	42.7	42.62	42.55	42.43	42.55
16-Jul-21	43.84	42.46	42.12	41.99	41.87	41.77	41.68	41.58	41.77
17-Sep-21	43.03	41.97	41.63	41.49	1.36	41.24	41.14	40.99	40.94
21-Jan-22	42.19	41.38	41.17	41.1	41.03	40.97	40.91	40.80	40.6
17-Jun-22	41.42	40.84	40.71	40.67	40.64	40.62	40.60	40.55	40.46
16-Sep-22	40.61	40.21	40.14	40.12	40.11	40.1	40.1	40.09	40.06
20-Jan-23	39.63	39.41	39.39	39.4	39.41	39.43	39.45	39.5	39.60
17-Mar-23	40.06	39.97	40	40.02	40.05	40.05	40.13	40.22	40.39

TABLE 3.2: Input data points for the implied volatility surface in function of strike and maturity. In the first row, values of strikes in percentage; in the first column, corresponding maturities.

3.2 Numerical results for Heston model

In this section we present the option pricing and Greeks results under the Heston model. We conclude the section with error estimates.

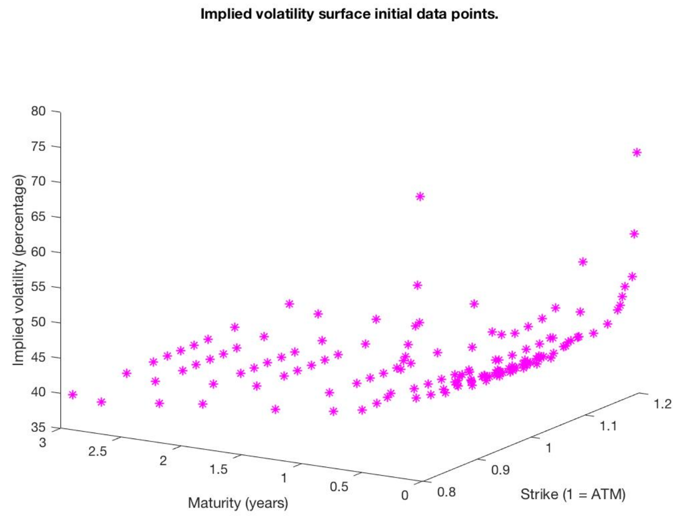


FIGURE 3.14: Input data points for the implied volatility surface.

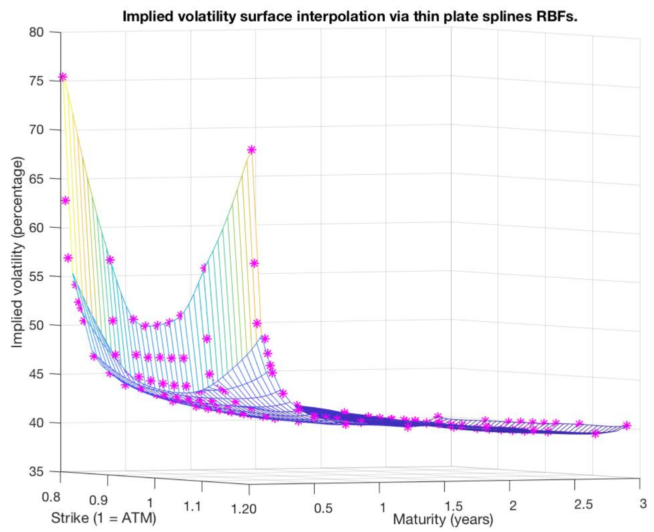
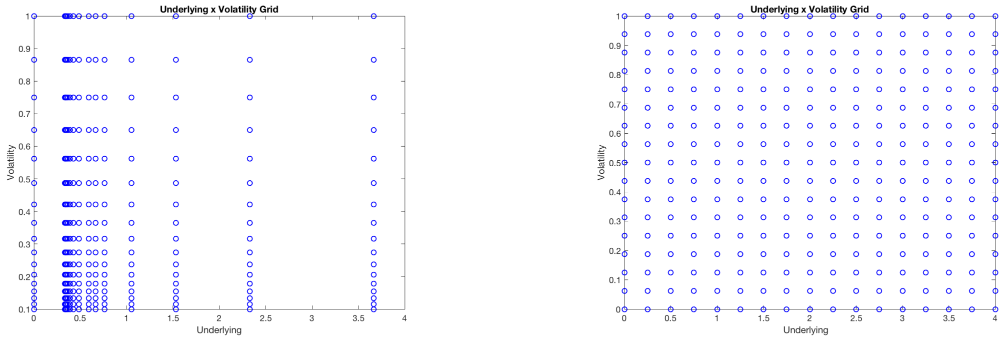


FIGURE 3.15: Resulting interpolated volatility matrix via RBFs.

3.2.1 Option pricing

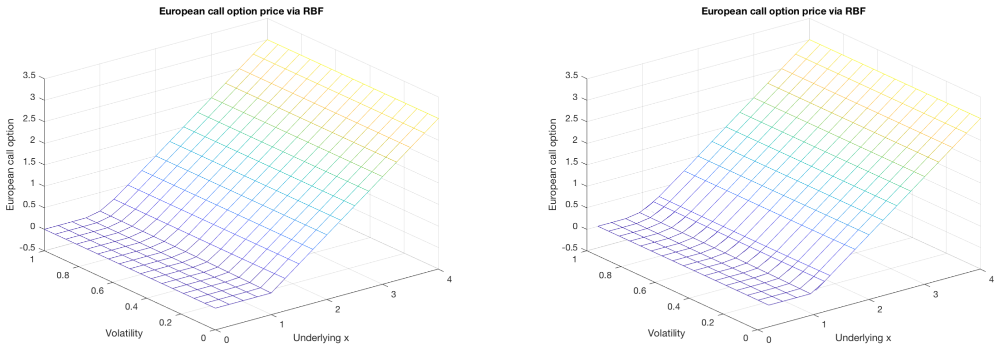
In all experiments we use multiquadrics as radial basis functions: we opted for a common type of RBF which is also easy to implement together with its derivatives. Multiquadrics are not in general the optimal choice for instability reasons. However, by also applying PUM, results were already satisfying for a selected range of values for the shape parameter. We used market data as inputs.
In Figure (3.16) we show the main examples of grid we worked with: a classical equidistant grid and a non-uniform grid, with a greater density of points around the critical strike region. In Figure (3.17) we show the prices of a European call option with maturity of half a year, risk-free parameter $r = 0.03$, $\rho = -0.05$, $k = 2$, $\sigma = 0.25$ and $\theta = 0.0225$, for two different grids.



(A) Non-uniform Underlying per Volatility grid.

(B) Equidistant Underlying per Volatility grid.

FIGURE 3.16: Example of two different grids: in Figure (3.16a) the distribution of points is denser around the strike region; Figure (3.16b) shows an uniform grid.



(A) Price calculated using an uniform grid.

(B) Price calculated using a non-uniform grid.

FIGURE 3.17: Price of a European call option via RBF resolution of Heston PDE with different grids.

We conclude from our pricing experiments that an uniform grid is sufficient to estimate an accurate enough price for a European call option. This choice also allows for a direct comparison of RBF methods with mesh-dependent ones, without having to increase the complexity of calculation and the computation time of, for example, a finite-difference scheme. Furthermore, we noticed that by increasing the density of the points around the strike region we also increase the chance of creating interpolation matrices which are badly scaled and ill-conditioned: this resulted in the failure of RBFs.

In pictures (3.18) and (3.19) we show the prices of a European option calculated with both methods for the same inputs and grid. In picture (3.20) we plotted the absolute difference between RBF and ADI prices.

In particular, the prices of a European vanilla option with volatility $\nu = 1$ and underlying value $S = 4$ units of currency, which we chose as limit values of our grid, are reported in table (3.3).

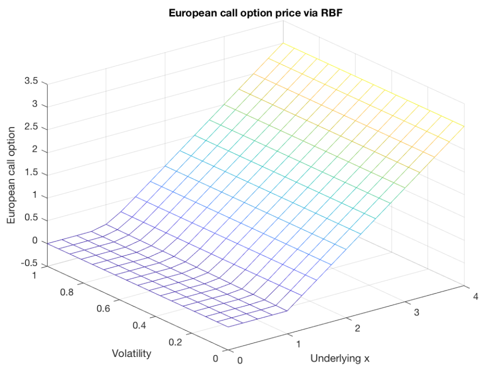


FIGURE 3.18: Price of a European option modeled with Heston model via RBF methods.

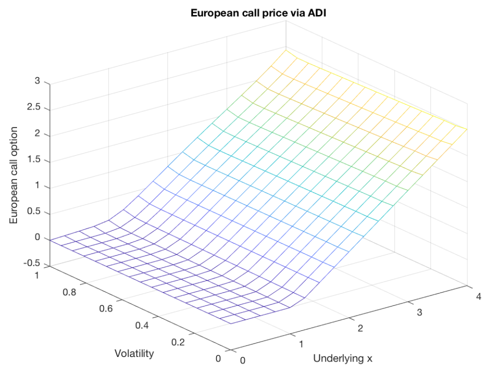


FIGURE 3.19: Price of a European option modeled with Heston model via ADI scheme.

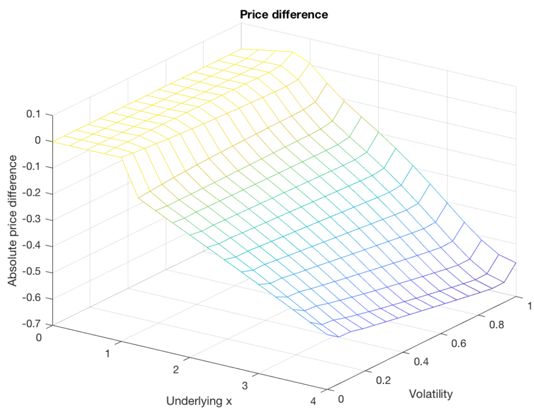


FIGURE 3.20: Absolute difference between RBF and ADI prices.

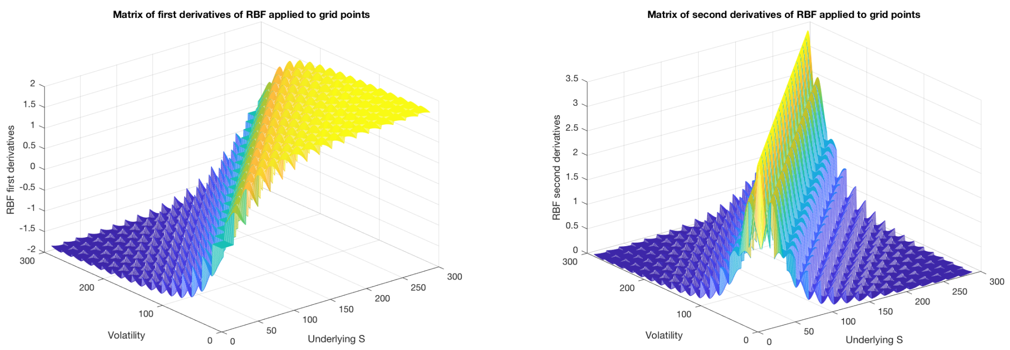
TABLE 3.3: European call option prices in USD calculated with different methods.

European call option price		
RBF Method	ADI Method	Analytical formula
3.0013	3.2251	3.0013
3.0021	3.0056	3.0022
2.1561	2.1439	2.1566

For the same input grid, RBF almost exactly matches the price generated by Heston analytical formula on most cases, up to an order of 10^{-05} in the worst cases. On the other hand, ADI is not able to equally perform. Differences seem to increase with higher underlying and volatility values: absolute differences between ADI and the analytical formula may reach the order of 10^{-02}. Furthermore, we noticed RBF produce results closer to the analytical formula in case of longer maturities of the option. Therefore, we can conclude RBF methods provide a more precise and efficient tool to solve Heston PDE.

3.2.2 Greeks calculation

We will now present the results for Delta and Gamma profiles. In Figure (3.21) we show the surfaces created by applying the derivatives of RBFs to the grid points: by multiplying for the lambda coefficients at today's time $t = 0$, we can finally smooth out the surfaces and obtain the Delta and Gamma of the option.
In Figure (3.22) we show the Delta profile of the same European call option, plotted against the volatility and underlying values. Eventually, in Figure (3.23) we present the Gamma profile of said option. Resulting surfaces are reasonably smooth, and we were able to obtain them without slowing down the computation of the pricing algorithm: this is a clear improvement with respect to mesh-dependent schemes, for which we are required to increase the computation time to obtain a not so smooth Delta or Gamma profile.



(A) First RBF derivatives surface. (B) Second RBF derivatives surface.

FIGURE 3.21: Surfaces representing the first and second derivatives RBF matrices with respect to the underlying, applied at the grid points.

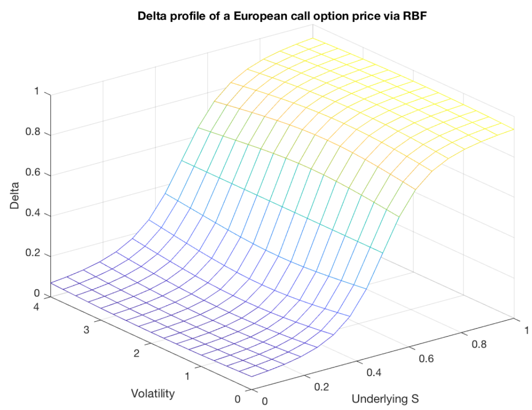


FIGURE 3.22: Delta profile of a European call option calculated by RBF methods in a Heston model framework.

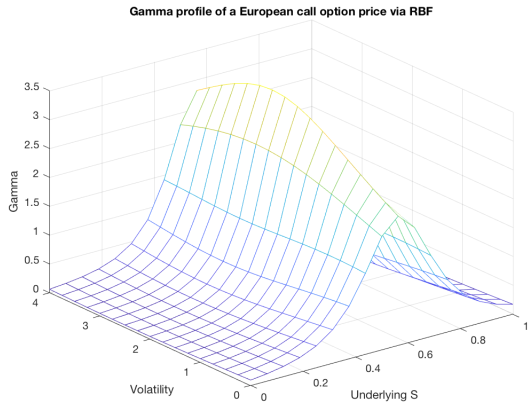


FIGURE 3.23: Gamma profile of a European call option calculated by RBF methods in a Heston model framework.

3.2.3 Error estimation

We define the errors respectively committed by RBF and ADI in calculating the solution to the Heston model as:

$$\Delta u_{max}^{RBF} = max_K |u_{RBF}(S_0, v_0, 0) - \bar{u}(S_0, v_0, 0)|, \tag{3.4}$$

$$\Delta u_{max}^{ADI} = max_K |u_{ADI}(S_0, v_0, 0) - \bar{u}(S_0, v_0, 0)| \tag{3.5}$$

where $\bar{u}(S_0, v_0, 0)$ denotes the solution computed by applying the ADI method with a greater number of steps. We calculate the error as a function of time for both methods. As shown in figure (3.24), ADI seems to have a faster convergence to a low error, while RBF solution presents a larger error at the beginning of the computation and later starts to lower to a reasonable error value, outputting a solution faster than ADI.

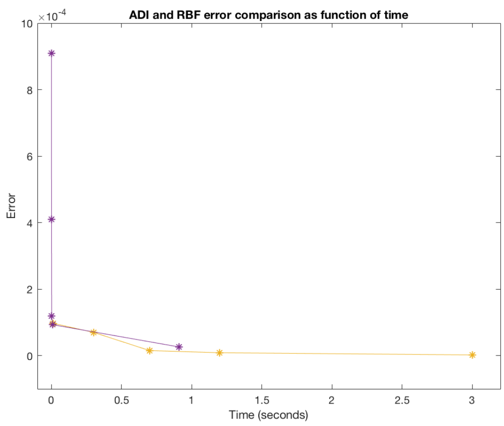


FIGURE 3.24: Error results for a European option under the Heston model.

3.3 Operational framework

In many application scenarios, processes make use of huge amounts of interrelated data: this explains the necessity of techniques for their classification and managing. Internet of Things (IoT) frameworks are very suitable to this kind of contexts for several reasons, in particular: the diffusion of sophisticated tools such as smart phones, tablets and smart watches; the possibility of real time data; efficient communication models among devices, e.g. Device-to-Device Communications, i.e. two or more devices that directly connect and communicate between one another; Device-to-Cloud Communications, where the IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic; and Device-to-Gateway Model, where there is an application software operating on a local gateway device, which acts as an intermediary between the device and the cloud service and provides security and other functionality such as data or protocol translation.
In banking context, IoT applications are able to improve underwriting processes for several purposes: obtaining more information of goods; monitoring the condition of different assets market; helping traders to choose the best opportunity. In particular, data retrieving, analysis e management are usually known as complex task in financial contexts. In an IoT system data-flow, processes represent the knowledge base used in mathematical models for credits and financial products. Several sources such as distributed database systems, portals and local information are generally used as input of inferring models.
In this section we describe an overview of software tools, methodologies and strategies in real data-flow system. In (3.3.1) we describe an example of a real data-flow system; in (3.3.2) we focus on software commonly used and we eventually present a simple parallelization experiment in (3.3.3). We will mainly refer to our work in [**parallelJournal**] and [**damis**].

3.3.1 Databases and data flow

In Finance, data can be involved in a complex and long process which allows the financial institution to properly treat and make advantage of them [30]. Data losses, misinterpretation and optimization are one of the main issues financial - and non - institutions must face.
We can wrap up the data-flow process in three main phases:

- **Data retrieving**: the financial institution retrieves data from more than one source, often causing operational risk.
- **Data analysis and management**: data are extracted from the database to be used for analysis purposes.
- **Data reporting**: data are transferred again to the main database and/or externally reported.

This general and simplified scheme is represented with more details in figure (3.25).

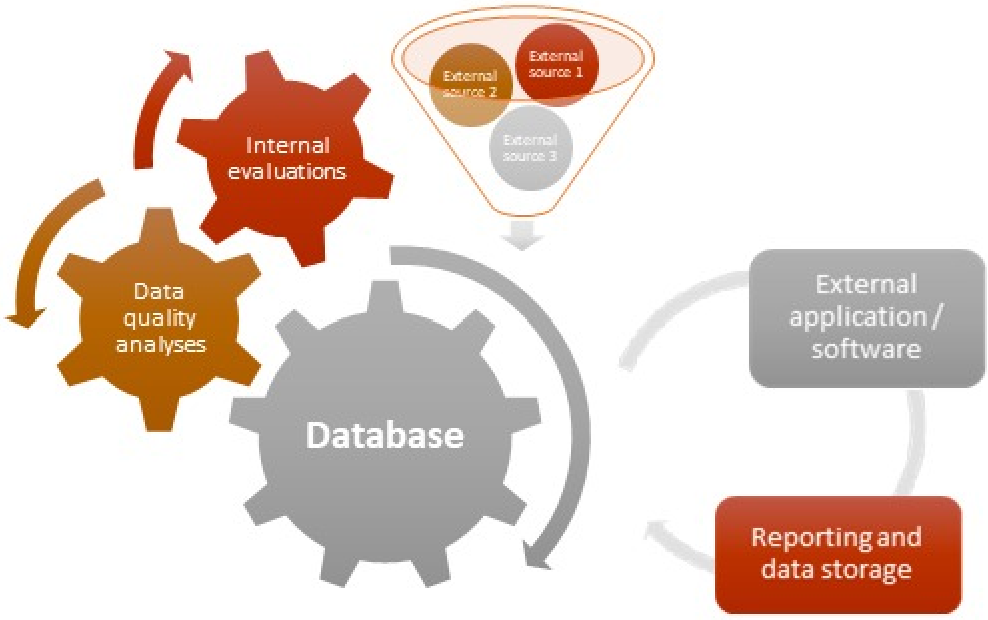


FIGURE 3.25: Data-flow example.

The ideal situation for a financial institution would be gathering all data from only one database. Nowadays, Business Intelligence is taking care of this aspect and is facing the tough challenge of joining old databases in a new, functional one.
Typically, given the huge size of banks and their step by step adjustment to new software and technologies, it is still hard to combine all data, particularly when treating historical ones. This is the reason why also - if not especially - huge financial institutions are still working with more than one database, generating overlapping data which in general do never exactly match.
We will here provide an example of how databases containing data for a Risk Management department could be structured. We can picture a database for credit risk data and a second database for market risk data retrieval.

- **Credit Risk database.**

  This database contains all data for credit risk purposes. Data are gathered, filtered, checked and finally used as input for credit risk models. Eventually, data are reported to other departments or saved in the main database, such that it will be possible to use them again. Given the different goals, various libraries are available in the database:

  1. A first library would contain historical data. It could be a copy of an old database, or just a static collection of tables ready to be read and used.
  2. A second library contains external sources data, that are uploaded and remain static or/and are continuously updated.

3. Some calculations can be performed directly in the database, so a library will be used to store all the calculators and their results. These data are read to be analyzed.
4. When analyses are performed, results are stored in a particular library that everyone can access from their local environment.
5. Since calculations and analyses may take a long time, it is useful to produce tables in intermediate steps to check the process status. Given the large amount of resulting data, it is better to store these intermediate results in a local library which will be deleted at the end of the process.

In each library, tables with all data can be selected and filtered or exported. We can assume data are uploaded at different time intervals: new loans can be added daily; calculators perform monthly; etc.

- **Market Risk database.** A Market Risk database will contain more data coming from external sources, often not free. Data are used to monitor traders and bank portfolios, develop and validate market risk models, and price financial instruments with advanced techniques.

  Data will be provided by:

  1. External sources:
     a. Mainly Bloomberg, with a specific add-in that allows to download market risk data directly to the database, minimizing operational risk;
     b. Markit, Superderivatives, TriOptima etc., id est further market data providers similar to Bloomberg. Excel files can be manually downloaded from their websites and later properly formatted and loaded into the database.
  2. Internal sources: traders, who upload data regarding every new product they sell or buy. Usually this process is manually done, producing operational risk. Market risk managers or Product Control analysts check data quality of inserted data.

3.3.2 Software

Data are processed to be filtered, checked, and eventually used as input for models or analyses. There are different types of software that can be chosen for these tasks. We will present the most common in financial world.

1. **Data Management Tools.**
   To manage databases and perform filters, data quality check or to build basic calculators, financial institutions use SQL Management Studio or/and SAS.

a. SAS. Useful for data storage and filtering, but commonly preferred because of its predefined models for Statistics and Data Analysis. SQL procedure allows to use SQL syntax; SAS programming language is not easy to fully understand. User-friendly, it can be used even without knowing how to program. Not fast for modelling as other programming languages could be.

b. SQL Management Studio. Specialized in data storage and filtering, it can easily produce reports. It is free-source and very fast. The syntax is clear and easy to learn.

2. **Data Analysis and Modelling Tools.**
Various programming languages can be chosen for modelling, depending on the specific requirements. Some languages are faster than others, but they take time to efficiently program.

a. R/RStudio. The most common for data analysis and Statistics. Free-source, can be adjusted to become object-oriented. Usually fast, it can be faster thanks to parallel computing techniques. $C++$ and Fortran programs can be easily ran in R. It works better with *csv* files. Easily linked to SQL.

b. MATLAB. Not free-source, but more reliable. It offers packages for parallel computing, global optimization, financial modelling. Already fast and with a user-friendly IDE. Not object-oriented. Easily linked to Excel and SQL.

c. Python. Free-source, it is fast and very precise. Good for heavy computational tasks.

d. C++ and Fortran. Faster than other languages, but they require more time for coding. Object-oriented. Definitely more precise in computation. They are preferred for heavy computational tasks.

Financial institutions usually prefer to use programs/types of software they have to pay for, since this allows them not to take responsibility on the results of the model. Indeed, if a particular package of a free-source software contains a bug, it will be the bank's responsibility to justify the mistake.
Eventually, data are ready to be reported to an external institution or to another department. Excel files are the easiest way to send data and tables, but more sophisticated software can be used. Data-flow can be automatized so that the operational risk of copying and pasting in excel disappears. E.g., reports can be produced directly with RStudio, which performs calculations on input data and then returns dataframes or datatables that can be already passed to the report.

Packages such as *Rmarkdown* and *Knitr* allow to use only one function to create a report, just passing the input parameters to the function. The process is fully described in figure (3.26): an *.Rmd* file with markdown text (similar to latex syntax) and R code chunks is passed to Knitr, that executes the chunks and creates a new markdown file with extension *.md*. Finally, the *.md* file is processed by *pandoc* that creates the final output: a Word file. Other types of output can be selected, such as a *pdf* or *htlm* file, opportunely choosing LaTeX or other document converters.

FIGURE 3.26: Rstudio: final output creation.

The purpose of data analysis and quality check in our example is to furnish input data for risk models: hence, models heavily depend on data retrieving. This concept is well shown in figure (3.27).

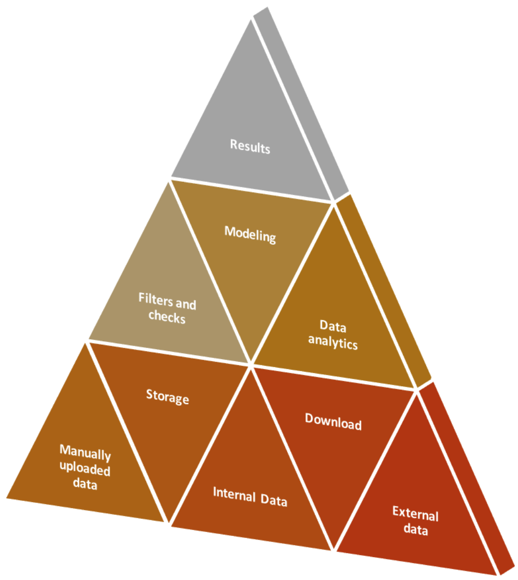


FIGURE 3.27: Data modelling: from data retrieval to resulting output.

3.3.3 Parallelization techniques

Several processes in Quantitative Finance require the usage of high-performance computing and efficient software. Different programming languages, modeling tools and fast calculators are employed for better performances.
Parallel Computing techniques may be useful in different kind of issues:

- Data retrieval: split the incoming amount of data in more buckets to channel to the database to obtain a low latency with cluster machines;

- Monte Carlo simulation: chunks of the code can be parallelized since they are independent in the calculation. Up to a number of simulations in time, a bucket can be created. Depending on heavier products, there will be uneven blocks;

- Derivatives Pricing: data parallel programming languages such as High Performance Fortran for advanced applications (HPF+) with extensions for clusters of symmetric multiprocessing (SMPs) have been employed in derivatives pricing, when the algorithms are of lattice type.

Massive daily calculations require the usage of specific tools, such as Graphics Processing Unites (GPUs) (3.28):

- Monte Carlo VaR: all the products in the portfolio are re-priced with stressed values in the future time. All the simulations are sorted, which is also an expensive operation.

- XVA calculations: the main issue where GPU computing makes the real difference. CVA, DVA, etc. involve the simulation of at least 1000 point, daily and in future time until the contract maturity for each risk factor: $1000 \cdot 17600 \cdot 3$ points for a single contract of maturity 50y and dependent on 3 risk factors. This calculation is daily for all the products in the portfolio. Eventually, exposures are computed and collateral is modeled and added.

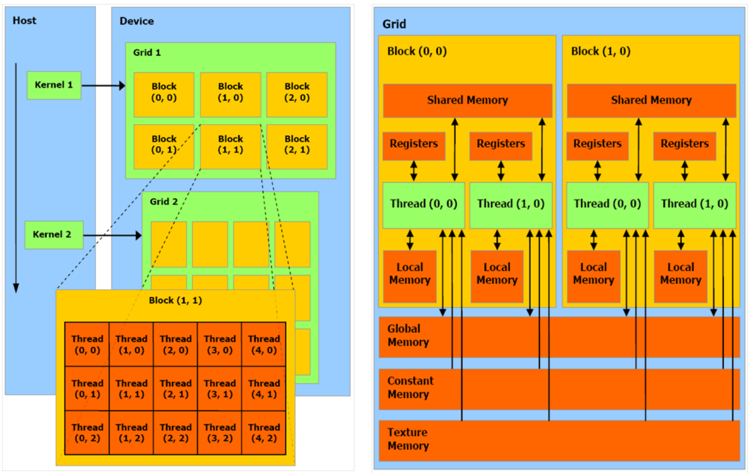


FIGURE 3.28: Gpu outline.

As an example of parallelization techniques, we will show a simple interest rate model implementation using parallel computing techniques in R: the one-factor Hull-White model under risk-neutral measure **[HW2]**.
We consider the filtered, **real-world probability** space $(\Omega, \mathcal{F}, \{\mathcal{F}_n\}, \mathcal{P})$ and the martingale measure Q, associated with a classical bank account numeraire $B(t)$. The One-factor Hull-White model is represented by the following equation:

$$dr(t) = (\theta(t) - ar(t))dt + \sigma dW(t). \tag{3.6}$$

where a - the mean reversion parameter - and σ - the volatility - are constants; $r(t)$ is the interest rate at time t; θ is the mean reversion speed at time t and W is a one-dimensional Wiener process.

The solution of this equation is provided by:

$$r(t) = E^{Q}[r(t)|\mathcal{F}_s] \pm \sqrt{Var^{Q}(r(t)|\mathcal{F}_s)} \cdot Z \tag{3.7}$$

$$E^{Q}[r(t)|\mathcal{F}_s] = r(s)e^{-a(t-s)} + \alpha(t) - \alpha(s)e^{-a(t-s)} \tag{3.8}$$

$$Var^{Q}(r(t)|\mathcal{F}_s) = \frac{\sigma^2}{2a}[1 - e^{-2a(t-s)}] \tag{3.9}$$

$$\alpha(t) = f(0,t) + \frac{\sigma^2}{2a^2}(1 - e^{-at})^2 \tag{3.10}$$

$$f(0,t) = -\frac{\partial lnP(0,t)}{\partial t} \tag{3.11}$$

where Z is a standard normal random variable, $f(0,t)$ is today's instantaneous forward rate for maturity t and $P(0,t)$ is today's discount bond price for maturity t.
To be able to use parallel computing in R, the package *parallel* is required. We use the command $n\_cores = detectCores()$ to calculate the number of available cores and then we subtract one to that number: we will create $n-1$ clusters.
To simulate this simple model, we retrieve all data from the database and from Bloomberg. We precisely need the forward rates and the bond prices. The mean reversion a and the volatility σ are given by a proper calibration. We choose EURIBOR 6M as yield curve and $NumSims = 10000$ as number of simulations.
he simulation is implemented in R language, on a multicore, x86-64 Windows machine. We can calculate $\alpha(t)$ in R, since it is still fast, and then export it with all the other inputs to the clusters through the function *clusterExport(cluster, "input_name", envir=environment())*. We export the function that calculates the rates and, at the same time, we run the function by using $clusterEvalQ(cluster, HW\_Rates())$.
We conclude that in this case the parallel technique speeds up the calculation, producing three different results, i.e. $10.000 \cdot 3 = 30.000$ simulations, in about 40 seconds. The same calculation in R without parallel computing takes approximately 20 seconds for only one result (10.000 simulations). Furthermore, calculations can be even faster if we include in the cluster the α function and the calculation of other inputs, since exporting large data to the clusters could take further time.
We present the results for the calculated forward rates in (3.29). Given the spiky behavior of the forward rates, the simulated Hull White rates will replicate such behavior for the first days. We show Hull White rates plotted for the 100-th simulation in (3.30).

Eventually, we report in table (3.4) part of the results of the simulation. We can observe from the table how the results become more and more volatile as we go further in time.

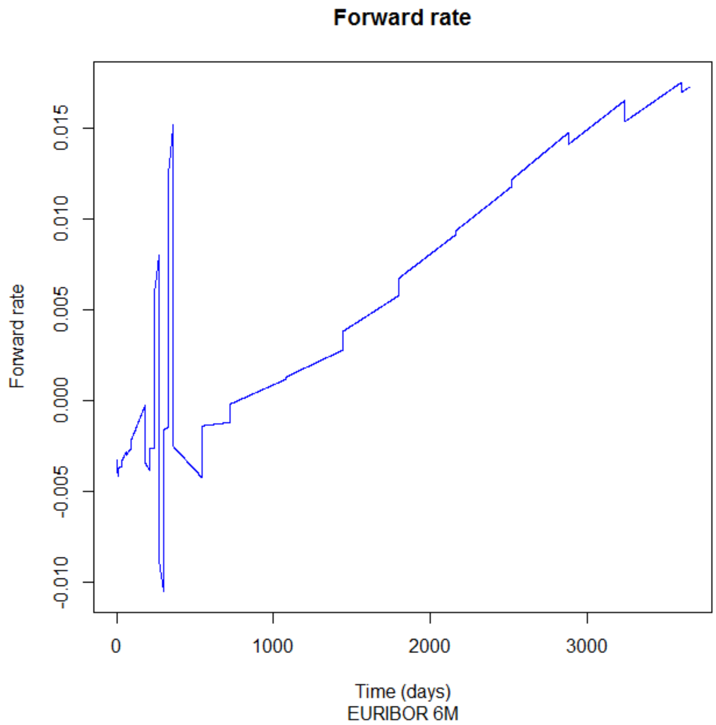


FIGURE 3.29: Forward rates.

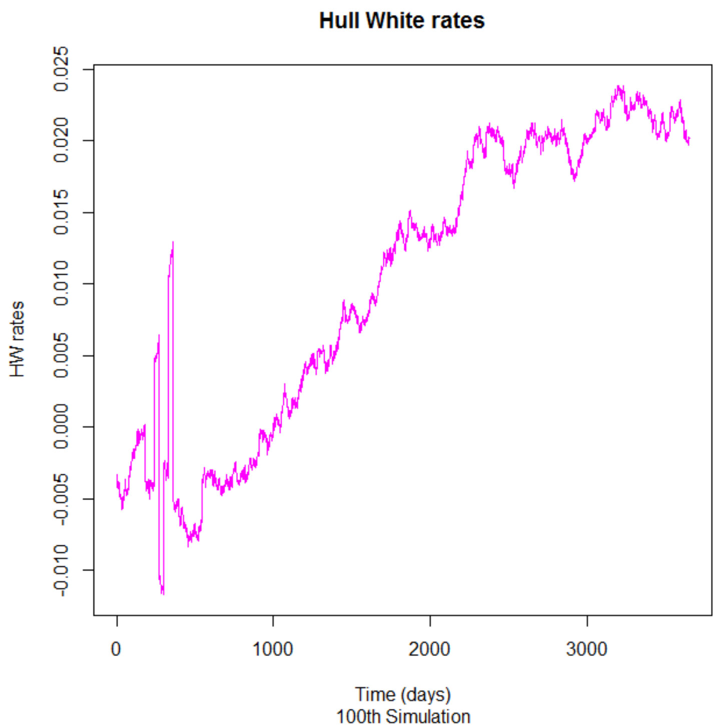


FIGURE 3.30: 100th Simulation of Hull White Rates.

TABLE 3.4: Hull White Rates simulation. We indicate by *n* the number of simulations.

Hull White Rates Simulation.					
Days	n = 1	n = 2	n = 500	n = 9999	n = 10000
1	-0.0033	-0.0032	-0.0033	-0.0033	-0.0033
10	-0.0041	-0.0039	-0.0033	-0.0031	-0.0032
100	-0.0039	-0.0002	-0.0012	-0.0059	-0.0039
1000	-0.0056	0.0059	0.0068	-0.0016	0.0055
2000	0.0077	0.0076	0.0103	0.0064	0.0192
3000	0.0142	0.0149	0.0198	0.0049	0.0315
3650	0.0168	0.0210	0.0063	0.0152	0.0374

Conclusions

In this thesis we have explored the usage of Radial Basis Functions numerical methods to solve mathematical problems in Finance, such as the pricing of financial derivatives and the calculations of Greeks. Our main contributions lie in extending such methods to the pricing of more complex financial derivatives, and in deriving the theoretical formulae to numerically compute the associated Greeks.

We started by applying RBFs to a standard Black-Scholes equation, and we evolved the PDE step by step by adapting the methods to the new hypotheses, dealing in particular with the Heston model. Furthermore, we compared the results to the most reknown techniques, such as Finite Difference Methods. We also reproposed an alternative Monte Carlo, namely the Particle Filtering numerical technique, and commented the results.

Overall, our method outperformed FDMs in both Black-Scholes' and Heston model's cases: resulting prices' plots seem to be smoother than the corresponding FDMs ones. Our results also show a gain in efficiency when choosing RBFs: we can derive the Greeks almost for free - i.e. without much more computational effort - with respect to the FDMs and Monte Carlo cases. Moreover, RBF-derived Greeks are clearly smoother. Eventually, we have also set up the framework for eventual future research by showing how to construct a RBF interpolation of an implied volatility surface. Further work could consist in extending the idea to local or stochastic volatility surfaces.

Appendix A

Stability and Trade-off principles for Radial Basis Functions

In this appendix we are going to summarize the main properties of radial basis functions. Being radial basis functions an approximation method, it is meaningful to study its condition number. The *condition number* of an interpolation method measures how much its output can vary for a small change in the input argument. By doing this, we can measure how sensitive the method is to changes or errors in the input, and how much the error in the input will modify the error in the output.
Let us consider the interpolation matrix A associated to a given problem, and which we have defined with entries $A_{ij} := \Phi(\mathbf{x}_i - \mathbf{x}_j)$ with $\mathbf{x}_i, \mathbf{x}_j$ points of the input set X. Its l_2*-condition number* $cond(A)$ is defined as:

$$cond(A) := ||A||_2||A^{-1}||_2 = \frac{\sigma_{max}}{\sigma_{min}}, \tag{A.1}$$

where σ_{max} and σ_{min} are the largest and smallest singular values of the interpolant matrix A. If the matrix A is also positive-definite, the condition number may be computed as:

$$cond(A) := \frac{\lambda_{max}}{\lambda_{min}}, \tag{A.2}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of A. Bounds can be easily found for λ_{max}. In fact, by using Gershgorin's theorem:

$$|\lambda_{max} - A_{ii}| \leq \sum_{j=1, j\neq i}^{N} |A_{ij}| \tag{A.3}$$

for some $i \in \{1, \ldots, N\}$. Hence, by applying the definition of the interpolant matrix A we have:

$$\lambda_{max} \leq N max_{i,j=1,\ldots,N}|A_{ij}| = N max_{\{\mathbf{x}_i, \mathbf{x}_j\}\in X}|\Phi(\mathbf{x}_i - \mathbf{x}_j)|. \tag{A.4}$$

Being Φ strictly positive definite, we obtain (see properties of strictly positive definite functions in chapter (2) [17]):

$$\lambda_{max} \leq N\Phi(\mathbf{0}). \tag{A.5}$$

This means λ_{max} will not grow too fast if the input data are decently distributed. Regarding λ_{min}, several papers have covered the subject [2], [39], [55]. In particular, Ward and Narkowich establish bounds in terms of separation distance - or, as sometimes it is referred to, the *packing radius* - q_X of the data points:

$$q_X := \frac{1}{2} min_{i \neq j} ||\mathbf{x}_i - \mathbf{x}_j||_2, \tag{A.6}$$

which we can imagine as the radius of the largest ball which we can draw around a data point without intersecting the balls centered in the remaining points. It can be actually calculated as half the minimum of the matrix of distances.
The dependence of the error on the separation distance q_X results in the so-called **first trade-off or uncertainty principle**, which consists in the inverse proportion between accuracy and stability of the radial basis function interpolation. This principle has led researchers to search for an *optimal* shape parameter ϵ which would result in the best achievable accuracy without deteriorating stability.
The **second trade-off or uncertainty principle** involves the shape parameter. Studies such as [12], [33], [59], [21] propose the Contour-Padé integration algorithm to compute multiquadrics with an extremely small value of ϵ in order to obtain maximum accuracy. However, such method is limited to small data sets.
The last but not least, **third trade-off or uncertainty principle** regards compactly supported functions. In case of stationary interpolation, we may apply methods that are numerically stable but are not efficient [46]: convergence is obtained at the cost of getting densely populated and/or ill conditioned interpolation matrices.

Appendix B

Catalog of RBFs with Derivatives

In this appendix we provide the main formulae for mentioned RBFs [60], [18].

Guassian RBF

This function is globally supported and strictly positive definite. It is C^∞ at the origin.

$$\phi(r) := e^{-(\epsilon r)^2}, \tag{B.1}$$

$$\frac{d}{dr}\phi(r) = -2\epsilon^2 r e^{-(\epsilon r)^2}, \tag{B.2}$$

$$\frac{d^2}{dr^2}\phi(r) = 2\epsilon^2(2(\epsilon r)^2 - 1). \tag{B.3}$$

Inverse Multiquadric RBF

This function is globally supported and strictly positive definite. It is C^∞ at the origin.

$$\phi(r) := \frac{1}{\sqrt{1+(\epsilon r)^2}}, \tag{B.4}$$

$$\frac{d}{dr}\phi(r) = -\frac{\epsilon^2 r}{1+(\epsilon r)^{2\frac{3}{2}}}, \tag{B.5}$$

$$\frac{d^2}{dr^2}\phi(r) = \epsilon^2\frac{2(\epsilon r)^2 - 1}{1+(\epsilon r)^{2\frac{5}{2}}}. \tag{B.6}$$

Linear Matérn RBF

This function is globally supported and strictly positive definite. It is only C^2 at the origin.

$$\phi(r) := e^{-\epsilon r}(1+\epsilon r), \tag{B.7}$$

$$\frac{d}{dr}\phi(r) = -\epsilon^2 r e^{-\epsilon r}, \tag{B.8}$$

$$\frac{d^2}{dr^2}\phi(r) = \epsilon^2 e^{-\epsilon r}(\epsilon r - 1). \tag{B.9}$$

Multiquadric RBF

This function is globally supported and strictly conditionally positive definite of order 1. It is C^∞ at the origin.

$$\phi(r) := \sqrt{1+(\epsilon r)^2}, \tag{B.10}$$

$$\frac{d}{dr}\phi(r) = \frac{\epsilon^2 r}{\sqrt{1+(\epsilon r)^2}}, \tag{B.11}$$

$$\frac{d^2}{dr^2}\phi(r) = \frac{\epsilon^2}{1+(\epsilon r)^{2\frac{3}{2}}}. \tag{B.12}$$

Thin Plate Splines

This function is globally supported and strictly conditionally positive definite of order 2. First and second derivatives' singularities at the origin are removable; third derivative's singularity is not.

$$\phi(r) := r^2 log(r), \tag{B.13}$$

$$\frac{d}{dr}\phi(r) = r(2log(r)+1), \tag{B.14}$$

$$\frac{d^2}{dr^2}\phi(r) = 2log(r)+3. \tag{B.15}$$

Wendland's function

This function is compactly supported and strictly positive definite in $(R)^3$. Not differentiable at the origin.

$$\phi(r) := max((1-\epsilon r)^2, 0). \tag{B.16}$$

List of Figures

List of Tables

Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance.

Prima facie, I would first like to thank my supervisor, Professor Emilia Di Lorenzo, whose guidance and support were crucial during these years. You provided me with trust and gifted me with the opportunity to start my journey. Your feedback pushed me to reach for a higher level of research.

I would like to thank my tutor, Professor Salvatore Cuomo, for the continuous assistance and perseverance in following my path. Our weekly meetings helped me sharpen my thinking and showed me the right direction to complete my dissertation. Your patience and optimism were determining factors in achieving my goals.

In addition, I would like to thank my boyfriend, Alfonso, who sacrificed uncountable weekends and holidays to spend with me in order to support my research. I would not be writing a dissertation without your infinite patience and unconditional faith in me. Finally, I could not have completed this dissertation without the support of my friends, Roberta and Costantino, Amalia, Giuseppe, for their sympathetic ear every time I needed to be cheered up.

Last but not least, I thank my grandparents: *Nonno* Sabato, who thought me the importance of studying and the love for Mathematics since I was a child; and *Nonna* Sabina, who used to tell me how important it is to think with my own head in order to become, one day, an independent woman.

Bibliography

[1] I. Babuška and M. Melenk. "The partition of unity method". In: *Int. J. Numer. Meths. Eng.* 40 (1997), pp. 727–758.

[2] K. Ball, N. Sivakumar, and J.D. Ward. "On the sensitivity of radial basis interpolation to minimal data separation distance". In: *Constructive Approximation* 8 (1992), pp. 401–426.

[3] T. Bjork. "Arbitrage Theory in Continuous Time". In: *Oxford University Press* (2004).

[4] F. Black and M. Scholes. "The Pricing of Options and Corporate Liabilities". In: *The Journal of Political Economy* 81.3 (1973), pp. 637–654.

[5] G.E.P. Box and N.R. Draper. "Empirical Model-Building and Response Surfaces". In: *Wiley* (1987).

[6] P. Boyle and D. Emanuel. "Discretely adjusted option hedges". In: *Journal of Financial Economics* 8 (1980), pp. 259–282.

[7] M.D. Buhmann. "Radial basis functions". In: *Acta Numerica* 9 (2000), pp. 1–38.

[8] R. Cavoretto and A. De Rossi. "On the search of the shape parameter in radial basis functions using univariate global optimization methods". In: *Journal of Global Optimizations* 2 (2019), pp. 1–23.

[9] S. De Marchi. "Lectures on radial basis functions". In: (2018), pp. 1–64.

[10] E. Derman and I. Kani. "Riding on a Smile". In: *Risk magazine* 7 (1994), pp. 32–39.

[11] J. Douglas. "Alternating direction methods for three space variables". In: *Numerische Mathematik* 4.1 (1962), pp. 41–63.

[12] T.A. Driscoll and B. Fornberg. "Interpolation in the limit of increasingly flat radial basis functions". In: *Computational Mathematics Applications* 43 (2002), pp. 413–422.

[13] D.J. Duffy. "Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach". In: *Wiley Finance* (2006).

[14] B Dupire. "A Unified Theory of Volatility". In: *Derivatives Pricing: The Classic Collection* (1996), pp. 185–196.

[15] B. Dupire. "Pricing with a Smile". In: *Risk magazine* 7 (1994), pp. 18–20.

[16] L.C. Evans. "Partial Differential Equations: Second Edition". In: *American Mathematical Society* (2010).

[17] G.E. Fasshauer. “Meshfree Approximation Methods with MATLAB”. In: *Interdisciplinary Mathematical Sciences* 6 (2007).

[18] G.E. Fasshauer. “Solving partial differential equations by collocation with radial basis functions”. In: *Surface Fitting and Multiresolution Method* 2 (1997), pp. 131–138.

[19] M.R. Fengler. “Arbitrage-Free Smoothing of the Implied Volatility Surface”. In: *SFB 649 Discussion Paper 2005-019, Economic Risk Berlin* (2005).

[20] B. Fornberg and N. Flyer. “Solving PDEs with radial basis functions”. In: *Acta Numerica* 24 (2015), pp. 215–258.

[21] B. Fornberg and G.B. Wright. “Scattered node compact finite differente type formulas generated from radial basis functions”. In: *Journal of Computational Physics* 212 (2006), pp. 99–123.

[22] C. Franke and R. Schaback. “Convergence orders of meshless collocation methods using radial basis functions”. In: *Adv. in Comput. Math.* 8 (1998), pp. 381–399.

[23] P. Glasserman. “Monte Carlo Methods in Financial Engineering”. In: *Stochastic Modelling and Applied Probability Series* 53 (2003).

[24] J. Glover and M.M. Ali. “Using radial basis functions to construct local volatility surfaces”. In: *Applied Mathematics and Computation* (2011), pp. 4834–4839.

[25] R.L. Harder and R.N. Desmarais. “Interpolation using surface splines”. In: *J. Aircraft* 9 (1972), pp. 189–191.

[26] R. Hardy. “Multiquadric equations of topography and other irregular surfaces”. In: *J. Geophys.* 76 (1971), pp. 1905–1915.

[27] S.L. Heston. “A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options”. In: *The Review of Financial Studies* 6 (1993), pp. 327–343.

[28] C. Homescu. “Implied Volatility Surface: Construction Methodologies and Characteristics”. In: *SSRN Electronic Journal* (2011).

[29] J. Huang and J.S. Pang. “Option pricing and linear complementarity”. In: *Journal of Computational Finance* 2 (1998), pp. 31–60.

[30] J. Hull and A.D. White. “Efficient procedures for valuing European and American path-dependent options”. In: *The Journal of Derivatives* 1.1 (1993), pp. 21–31.

[31] E.J. Kansa. “Multiquadrics: A scattered data approximation scheme with applications to computational fluid-dynamics, part I and part II”. In: *Computational Mathematics Application* 19 (1990), pp. 127–145, 147–161.

[32] P. Lancaster and K. Šalkauskas. “Surfaces generated by moving least squares methods”. In: *Math. Comp.* 37 (1981), pp. 141–158.

[33] E. Larsson and B. Fornberg. “Theoretical and Computational aspects of multivariate interpolation with increasingly flat radial basis functions”. In: *Computational Mathematics Applications* 49 (2005), pp. 103–130.

[34] H.E. Leland. “Option pricing and replication with transaction costs”. In: *Journal of Finance* 40 (1985), pp. 1283–1301.

[35] D. Madan. “Conic Portfolio Theory”. In: *SSRN paper* (2015).

[36] J. Meinguet. “Multivariate interpolation at arbitrary points made simple”. In: *Z. Angew. Math. Phys.* 30 (1979), pp. 292–304.

[37] G.J. Moridis and E.J. Kansa. “The Laplace transform multiquadric method: A highly accurate scheme for the numerical solution of linear partial differential equations”. In: *J. Appl. Sc. Comp.* 1 (1994), pp. 375–407.

[38] MS. Mukhametzhanov, A. De Rossi, and R. Cavoretto. “An Experimental Study of Univariate Global Optimization Algorithms for Finding the Shape Parameter in Radial Basis Functions”. In: *International Conference on Optimization and Applications* (2020), pp. 326–339.

[39] F.J. Narkowich and J.D. Ward. “Norm estimates for the inverses of a general class of scattered data radial basis function interpolation matrices”. In: *Journal of Approximation Theory* (1991), pp. 84–109.

[40] B.F. Nielsen, O. Skavhaug, and Tveito A. “Penalty methods for the numerical solution of American multi-asset option problems”. In: *Journal of Computational and Applied Mathematics* 222 (2008), pp. 3–16.

[41] Q. Niu. “No Arbitrage Conditions and Characters of Implied Volatility Surface: A Review for Implied Volatility Modelers”. In: *SSRN paper* (2016).

[42] A. Pascucci. “Calcolo stocastico per la finanza”. In: *Springer* (2007).

[43] A. Safdari Vaighani, A. Heryudono, and E. Larsson. “A radial basis function partition of unity collocation method for convection-diffusion equations”. In: *Journal of Scientific Computing* 64.2 (2015), p. 341.

[44] R. Schaback. “Creating surfaces from scattered data using radial basis functions”. In: *Mathematical Methods for Curves and Surfaces* (1995), pp. 477–496.

[45] R. Schaback. “Multivariate interpolation by polynomials and radial basis functions”. In: *Constr. Approx.* 21 (2005), pp. 293–317.

[46] R. Schaback. “On the efficiency of interpolation by radial basis functions”. In: *Surface fitting and Multiresolution Methods* (1997), pp. 309–318.

[47] D. Shepard. “A two dimensional interpolation function for irregularly spaced data”. In: *Proc. 23rd Nat. Conf. ACM* (1968), pp. 517–524.

[48] S. Shreve. “Stochastic Calculus for Finance II: Continuous-Time Models”. In: *Springer Finance* (2010).

[49] V. Shscherbakov and E. Larsson. "Radial Basis function partition of unity methods for pricing vanilla basket options". In: *Computers and Mathematics with Applications* 71 (2016), pp. 185–200.

[50] F. Sica, S. Cuomo, and V. Di Somma. "A Note on the Numerical Solution of Heston PDEs". In: *Ricerche di Matematica* (2019), pp. 1–8.

[51] F. Sica, S. Cuomo, and V. Di Somma. "Remarks on a financial inverse problem by means of Monte Carlo Methods". In: *Journal of Physics Conference Series* 904 (2017).

[52] F. Sica, S. Cuomo, and F. Piccialli. "RBF methods in a Stochastic Volatility framework for Greeks computation". In: *Journal of Computational and Applied Mathematics* 380 (2020).

[53] F. Sica, S. Cuomo, and G. Toraldo. "Greeks computation in the option pricing problem by means of RBF-PU methods". In: *Journal of Computational and Applied Mathematics* 376 (2020).

[54] F. Sica et al. "Implied volatility surface reconstruction by means of radial basis functions". In: *Working paper* (2021).

[55] J.D. Ward. "Least squares approximation using radial basis functions: an update". In: *Advances in Constructive Approximation: Vanderbilt 2003* (2004), pp. 499–508.

[56] H. Wendland. "Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree". In: *Adv. in Comput. Math.* 4 (1995), pp. 389–396.

[57] H. Wendland. "Scattered Data Approximation". In: *Cambridge University Press* (1995).

[58] P. Wilmott. "Derivatives: The Theory and Practice of Financial Engineering". In: *Wiley* (1998).

[59] G.B. Wright. "Radial Basis Function Interpolation: Numerical and Analytical Developments". In: *Ph.D. dissertation* (2003).

[60] Z. Wu. "Compactly supported positive definite radial basis functions". In: *Advances in Computational Mathematics* 4 (1995), pp. 283–292.