

**Louvain School of Management
and Nova School of Business & Economics**

Design of a pedagogical tool in Python to price and hedge European options

Project Master's Thesis submitted by
Michel Vanderhulst

With a view of getting the degrees:
International Master in Finance
Master 120 credits in Business Engineering, professional Focus

Supervisors:
Prof. Frédéric Vrins
Prof. Gonçalo Sommer Ribeiro

Academic Year *2020-2021*

Abstract

Pricing and hedging options are concepts that are difficult to grasp for many students. The fact that "the price" means the cost of setting-up a self-financing trading strategy is difficult to visualize for students and manual computations are both unpractical and very limiting.

In this thesis, we develop and implement Python web applications to price and hedge options. Based on user-chosen inputs, the applications output the delta-hedge strategies in plots and allow data exportation. In their development, robust code architecture and strong pedagogical explanations were emphasized.

The web applications can be found at: hedging-derivatives.herokuapp.com

This work main deliverable is the web applications. The present document's purpose is to support them: its contents are meant to be used when users are in need of supplementary mathematical details or explanations, or simply if they are interested in the apps' development and implementation.

Keywords: Quantitative finance, Derivatives replication strategies, Black-Scholes, Cox-Ross-Rubinstein, Software engineering, Web application development

Acknowledgments

This work could not have been possible without the help and support of various persons. For their patience and kindness, I am deeply grateful.

Firstly, this thesis main supervisor: Prof. Frédéric Vrans from LSM. He always answered all of my questions and emails very quickly. He provided me with constructive and numerous comments for both the web applications and this written report. I would also like to thank him for introducing me to the world of quantitative finance two years ago, when I started my Master's degree at the Louvain School of Management.

Secondly, this thesis second supervisor: Prof. Gonalo Sommer Ribeiro from Nova SBE. While his supervision came at the end of this work due to the Double Master's Degree, his positivity regarding my work encouraged me to go the distance the last few weeks. His remarks were also very precise and helpful.

To the following three friends, I cannot thank you enough. Nathan Simonis, for continuously giving me feedback and always pushing me forward to do my very best. Niels Lücking, for the times my programming knowledge lacked and you helped debugging. Aurélien Lambert, for having made our Double Master's Degree experience in Portugal such an academic and social success and always motivating me to work on this thesis.

Finally, I think of my parents. Their education and patience led me on these past five years and kept me going forward. Thank you.

Contents

Abstract	I
Acknowledgments	II
List of Figures	V
List of Tables	VI
List of Symbols	VI
1 Problem specifications and design choices	1
1.1 The pedagogical problem	1
1.2 Desired tool characteristics	2
1.2.1 Usability	2
1.2.2 Robustness	2
1.2.3 Deployability, maintainability and extendibility	3
1.3 Design choices	4
1.3.1 Tool theoretical model	4
1.3.2 Programming language of use	7
1.3.3 Defining thesis' deliverables objectives	9
2 Theory of derivatives pricing	10
2.1 Derivatives payoffs	10
2.2 Arbitrage-free pricing & risk-neutral valuation	12
2.2.1 No-arbitrage price	12
2.2.2 Risk-neutral valuation	13
2.3 Standard pricing models	14
2.3.1 Cox-Ross-Rubinstein model	15
2.3.2 Black-Scholes-Merton model	16
2.4 Risk-neutral pricing formulas	17
2.5 Replication strategy	19
3 Development and deployment of a web-based application	22
3.1 App development framework	22
3.1.1 Dash library	22
3.1.2 Heroku cloud platform	23

3.1.3	Git version control system	23
3.1.4	Workflow	24
3.2	Create & Deploy your own app: step by step	24
3.2.1	Software requirements	25
3.2.2	Configuration	25
3.2.3	Creating the app	27
3.2.4	Deploying the app	27
3.2.5	Maintaining and extending the app	28
3.3	App legal aspects	28
4	Presentation of created web applications	30
4.1	App code architecture	30
4.2	Apps functionalities	31
4.2.1	Example 1: European option CRR web app	32
4.2.2	Example 2: Exchange option BSM web app	34
5	Future work	37
5.1	Design	37
5.1.1	Application appearance	37
5.1.2	Design responsiveness	37
5.2	Performance	37
5.3	Extensions	38
6	Conclusion	39
7	References	40
8	Appendices	43
8.1	Formal definition of risk-neutral pricing	43
8.2	CRR pricing formula through replicating portfolio	44
8.3	CRR pricing formula through risk-neutral	45
8.4	CRR option pricing examples	48
8.5	CRR option hedging examples	51
8.6	BSM European option risk-neutral price full derivation	57
8.7	BSM European option Δ full derivation	63
8.8	BSM European option replicating strategy	66
8.9	BSM Exchange option risk-neutral price full derivation	67
8.10	BSM Exchange option replication strategy	75

8.11	Permissive MIT license used in the applications	77
8.12	Example of generated Excel file in European option web app	78
8.13	Example of generated Excel file in Margrabe option web app	79
8.14	Example of error handling in European option CRR web app	80

List of Figures

1	Tool objectives, desired characteristics and design choices	3
2	General theoretical model map	5
3	CRR: 1-period binomial tree	15
4	Continuous time European option replicating strategy	21
5	App development workflow	24
6	Git identity set-up CMD commands	26
7	Heroku login CMD command	26
8	Updating application example	28
9	App code architecture	30
10	Starting screen of European option (CRR) app	32
11	Input sub-tab of the European option (CRR) web app	33
12	Exchange option (BSM) app	34
13	Input sub-tab of the Margrabe option (BSM) web app	35
14	CRR: 1-period tree for stock price, derivative price and cash account	44
15	European option CRR: 3-period stock tree	48
16	European option CRR: 3-period option pricing tree	48
17	Asian option CRR: 3-period stock/sum tree	49
18	Asian option CRR: 3-period option pricing tree	49
20	European option CRR: (left) Number of shares to hold Δ & (right) Equity account trees	51
21	European option CRR: Cash account tree	52
22	European option CRR: Replicating Portfolio tree	53
24	Asian option CRR: (left) Number of shares to hold Δ & (right) Equity account trees	54
25	Asian option CRR: Cash account tree	55
26	Asian option CRR: Replicating Portfolio tree	56
27	Example of Excel file for European option web app	78
28	Example of Excel file for Margrabe web app	79
29	European option web app negative stock price and tree periods error handling	80

List of Tables

1	General model structuring and design	4
2	Local vs cloud software	6
3	TIOBE Index top 10 programming languages	8
4	Arbitrage opportunity example	13
5	All software required to create your own application	25

List of Symbols

S_t	Spot price at time t
K	Strike or exercise price
T	Maturity of option
$\psi(\cdot)$	Option payoff function
V_t	Derivative price at time t
u	Up factor in Cox-Ross-Rubinstein model
d	Down factor in Cox-Ross-Rubinstein model
$\mathbb{P}$	Historical probability measure
$\mathbb{Q}$	Risk-neutral probability measure
$\hat{p}$	Risk-neutral probability of going up in Cox-Ross-Rubinstein model
$\hat{q}$	Risk-neutral probability of going down in Cox-Ross-Rubinstein model
r	Risk-free rate
δ	Discretization step
μ	Drift of the underlying asset
σ	Volatility of the underlying asset
Π_t	Replicating portfolio at time t
Δ_t	Number of shares to hold at time t
W_t	Brownian motion at time t
β_t	Continuously compounded discount rate at time t

1 Problem specifications and design choices

1.1 The pedagogical problem

For finance students, the fact that the price of an option actually corresponds the cost of a self-financing replication strategy is often difficult to visualize and assimilate. For their teachers, the problem is reversed, as transmitting this concept is just as difficult.

Indeed, for students, being told that under some assumptions it is possible to create a position (based on shares of the underlying and cash borrowed at the risk-free rate) that will replicate the cash flows of an option whatever the stock trajectory is no different than hearing a magic formula.

Mathematical proofs on the topic, frequently quite detailed, prove on the contrary to be not very enlightening for students. Students often need to "see that it works" to believe it, but computations of such trading strategies can be quite lengthy and tedious.

This thesis' purpose is to address this problem through an open-source pedagogical tool, which based on inputs would automatically compute and display a visual of the strategy replicating the option's price, therefore proving its arbitrage-free characteristic.

Additionally, it is both the author's and supervisor's wish for this tool to be extendable, i.e. allowing future students to implement other replicable options and assets, or even functionalities themselves.

The tool's two main **objectives** are thus:

1. Its pedagogical vocation
2. Its extendibility

To ensure the objectives, the tool's **desired characteristics** were defined. The manner in which these characteristics were implemented through **design choices** led to the tool's various functionalities and limitations. The following subsections will thus develop both characteristics and corresponding design choices.

1.2 Desired tool characteristics

We start by selecting general software characteristics relevant to our needs. The following were simultaneously considered: **usability, robustness, deployability, maintainability and extendibility** (Thakur, 2019) (Richards & Ford, 2020).

Below, we present these general software characteristics and introduce the design choices taken to ensure them in the tool. Their exact methodology is given later on.

1.2.1 Usability

Referring to the ease with which the tool can be used by all types of users (Thakur, 2019), this characteristic is crucial to the tool's **pedagogical objective**.

Indeed, the financial concept behind this thesis is in itself complicated enough. Adding another layer of difficulty, i.e., the tool's usage, would obviously have a negative effect on the users' experience and financial understanding.

To account for the tool's usability, we considered the following designs: **combining a clear theoretical model for the tool with an attractive and modern graphical user interface (GUI)**.

The goal of the theoretical model is to properly define the tool's contents, input and output and flow of information. Combined with an attractive GUI would ensure the user's experience (Richards & Ford, 2020) and thus in our case the understanding of the financial concept.

1.2.2 Robustness

This characteristic refers to the the software's ability to "keep an acceptable behavior" even if it is fed with invalid data, extreme inputs or simply if unexpected errors happen (Fernandez, Mounier, & Pachon, 2005) (Thakur, 2019). Shortly put, robust code is code that handles all types of failures: incorrect, incomplete or unexpected input and exceptions.

In our case, this is linked to thesis' **pedagogical objective**: the number of possible input to options and their replication strategies is high and and users might select extreme input values. Indeed, given that users will be looking to improve their understanding, problematic input are to be expected. This illustrates the importance of robust code.

To account for it, we considered **extensive error handling for all the possible inputs of the derivatives and following software engineering common practices.**

1.2.3 Deployability, maintainability and extendibility

Intrinsically linked, these three characteristics refer respectively to:

- Deployability: the ease with which the tool goes from theoretical development to practical code implementation
- Maintainability: once the work is finished, modifications might be required to the software. Maintainability thus refers to the ease with which one can correct errors or maintain software up-to-date.
- Extendibility: as mentioned in the previous subsection, future students might implement other replicable options or assets. Extensions based on the replicability of this thesis' implementation methodology is thus crucial, and extendibility refers to the ease of doing so.

To account for them, we take three design choices: **a clear and solid code architecture, relevant choice of programming language and the writing of a step-by-step tool development methodology for future students.**

Figure 1 summarizes the tool objectives, desired characteristics and design choices.

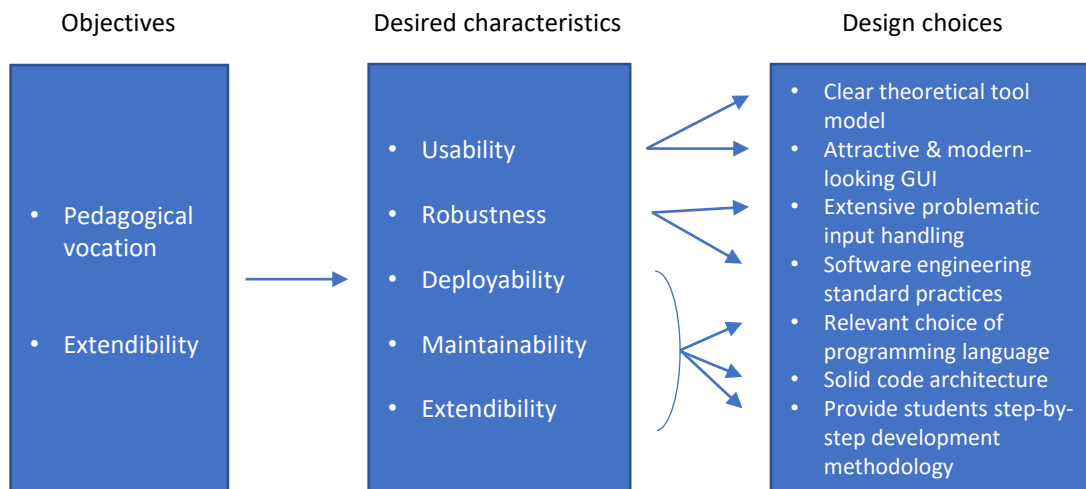


Figure 1: Tool objectives, desired characteristics and design choices
Source: taken and adapted from (Thakur, 2019) (Richards & Ford, 2020)

1.3 Design choices

This subsection further develops the design choices presented earlier. We start by defining a theoretical model for the tool. Then, we justify our choice of programming language and finally we define the goals of each of this thesis' deliverables.

1.3.1 Tool theoretical model

In financial modelling, **defining a model starts with its structure** (Eça, 2020). Table 1 gives the steps in doing so.

Model structuring
1. Determine the model's purpose
2. Define the output
3. Identify the required data
4. Define flow and content of major sections
5. Map the layout

Table 1: General model structuring and design

Source: taken from (Eça, 2020)

These steps have high return on investment, given that the rest of the tool development are based on them. Doing them properly will maximize structure, design and time gains.

1. Determine the model's purpose: Explain pedagogically that the price of an option is the cost of a self-financing replication strategy.
2. Define the output: Visuals and the generated replicating strategy. The former corresponds to graphics showcasing the replicating portfolio and the option's price. The latter to exporting to Excel the cash flows and financial operations that replicate the option's price.
3. Identify the required data: Derivative's inputs given by the user.
4. Define flow and content of major sections:
 - Section 1: Presentation of tool and asset
 - Section 2: Mathematical model used to price asset
 - Section 3: Methodology of implementation and replication strategy construction
 - Section 4: Inputs

- Section 5: Plots

For consistency's purposes across all types of options, the flow is the same for all of them. Naturally, the contents are adapted to each asset's specifications.

5. Map the layout: The general map of the model is given in **Figure 2**. Starting with the blue parts, we find the input and output. These are the only part of the model visible to the users.

Then, in gray and black are the invisible parts, namely the model fixed assumptions, the data and the engine. The engine corresponds to the mathematical computations. Note that no preliminary data is fed to the engine, given that all computations are based on user inputs.

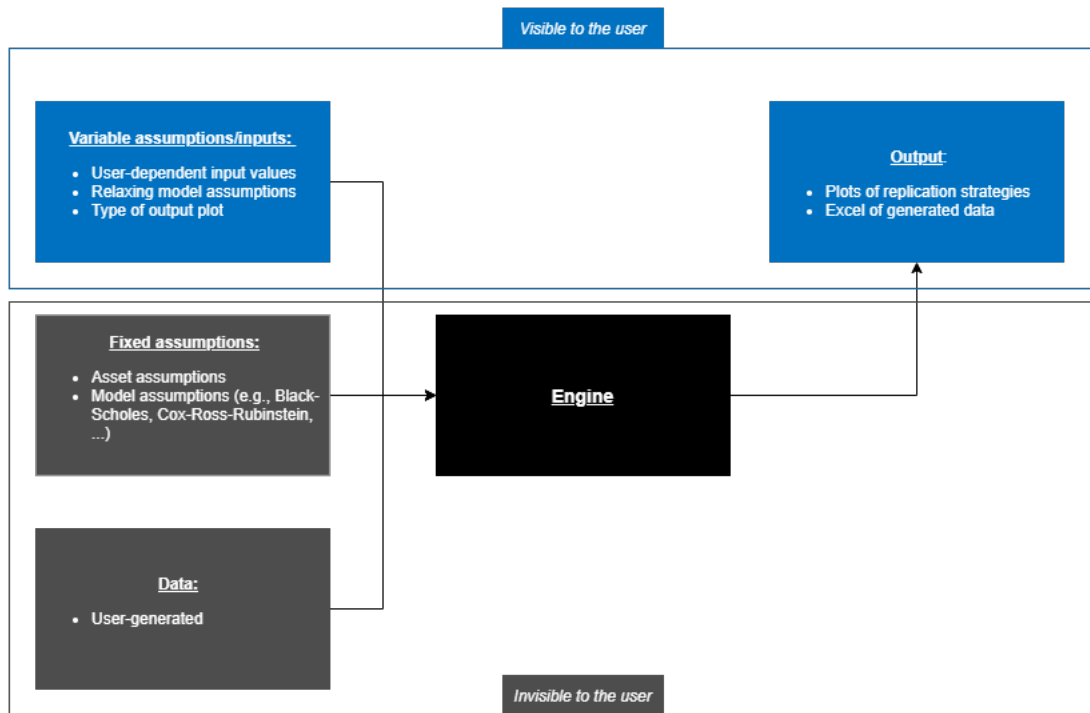


Figure 2: General theoretical model map

Source: adapted from (Eça, 2020)

After the general model structure comes the model construction (Eça, 2020). At this stage, the model must be formatted for the easiest usage based on the established structure.

To do so, two cases are considered simultaneously: local vs cloud software tools. Their main difference is on the set-up. Computer-based programs need to be installed on computers, and run locally. Web-based programs are hosted on web pages, and are delivered on demand, i.e. by accessing the internet page. The latter are also referred as Software as a Service (SaaS) (Morley & Parker, 2016).

For our decision, we use the already defined thesis objectives and design choices. A list of requirements is created based on them and then used to compare the pros and cons of both cases. We consider:

- Ease of accessibility (relevant to pedagogy, usability and deployability)
- User-friendliness & an attractive graphical user interface (relevant to pedagogy and usability)
- Easily editable and updatable (relevant to robustness, maintainability and extendibility)

Observe in **Table 2** the advantages and disadvantages of both cases (Morley & Parker, 2016).

Local	
Advantage	Disadvantage
Usable 24/7 without internet connection	Only runs locally
Unlimited customization	Operating-system dependent
Better performance	Manual and local updates
No web hosting fees	
Cloud	
Advantage	Disadvantage
Accessible 24/7 with internet access	Requires internet access
Operating system independence	Cyber security threats
Automatic and cross-users updates	Lower performance

Table 2: Local vs cloud software
Source: adapted from (Morley & Parker, 2016)

Based on the requirements,

- Ease of accessibility: cloud software is the better option, as most university students have access to both a computer and an internet connection. Furthermore, the internet link to the online tool would be provided by the teacher.
- User-friendliness & attractive graphical user interface: strictly in terms of customization, computer-based software is the better option. However, the operating system dependence requires high software engineering skills, which are out of the scope of this thesis.
- Easily editable and updatable: in the event of further extensions, or simply corrections to potential mistakes, cloud software is by far the better option. Indeed, only the application would need an update, saving the trouble of manual updates for all local installations.

From this analysis, a cloud application was settled on for development. The main take-away is a trade-off in customization for online access, ease of updates and operating system independence.

1.3.2 Programming language of use

To choose the programming language, we use the already defined thesis objectives, design choices and choice of cloud software. Based on them, four constraints were simultaneously considered: **the language popularity, programming object-orientation, existing web frameworks and usage in the financial industry.**

A popular programming language was immediately assumed to be required. The popularity is used as indicator of future usage, i.e., primordial for the tool's extendibility.

The TIOBE index was used as an proxy for the popularity of programming languages. Created by the TIOBE Company, the indicator is updated monthly. The indicator methodology is based on the number practitioners, courses and third party vendors, by using search engines such as Google, Bing, Yahoo!, Wikipedia, etc. Shortly put, it is a percentage of the number of queries per programming language covering most search engines (TIOBE, 2020a) (TIOBE, 2020b).

As of March 2020, the current ratings for the top 10 programming languages are observed in **Table 3** (TIOBE, 2020b):

Programming language	Rating
Java	17.78%
C	16.33%
Python	10.11%
C++	6.79%
C#	5.32%
Visual Basic	5.26%
JavaScript	2.05%
PHP	2.02%
R	1.26%
Groovy	0.25%

Table 3: TIOBE Index top 10 programming languages
Source: taken from (TIOBE, 2020b)

Then, more research on software engineering common practices showed that object-oriented programming (OOP) had to be considered to ensure the tool extendibility. Indeed, compared to traditional procedural-oriented programming, OOP promotes design structure, reusability of software and ease of code maintenance (Mössenböck, 1995). This allowed us to not consider the C language, as it is procedural-oriented.

Continuing with the usage in the financial industry, Python and R are the most sought-after according to the Financial Times (Murray, 2019). For the author or future finance students, it makes sense to chose accordingly to what the industry seeks. For this reason, Python made the most sense: "Python has established itself in the financial industry as a powerful programming language with an elaborate ecosystem of tools and libraries" (Hilpisch, 2015).

In addition to its strong industry presence, Python is also recommended altogether for multiple reasons:

- It is developer friendly with its clear syntax (Lei, Ma, & Tan, 2014)
- It is also open source, i.e. free in terms of license (Yan, 2017)
- It has thousands of libraries available with rich documentation (Hilpisch, 2015), such as Numpy, Pandas, Scipy and many others.

Finally, Python also has dozens web frameworks such as Django, Dash, CubicWeb, etc. (Python, 2020).

Python answering the four constraints, it was settled on to use it to develop an online application. The methodology and implementation followed are presented in a later section.

1.3.3 Defining thesis' deliverables objectives

Due to the nature of this thesis, the main deliverable is not this written document nor an empirical study but is rather a tool. It felt necessary to properly distinguish the goals of each:

1. The web applications: The applications are considered as the primary deliverable. Its goal, defined earlier in the theoretical model definition is to **explain pedagogically that the price of an asset is the cost of a self-financing replication strategy**.
2. The present written thesis: **Its goal is to support the web applications**. By support, it is meant that its contents are to be used when the students are in need of supplementary mathematical details, or if they are interested in the tool development. For this reason, the full mathematical derivations, proofs and examples will be given in the following section.

It is understood that both written thesis and tool go hand-in-hand, i.e. they are complementary.

2 Theory of derivatives pricing

This section reviews the general theoretical concepts of derivatives pricing. As mentioned earlier, this thesis has a pedagogical role: users of the web application that feel the need for additional mathematical details will go through this written report. Therefore, the explanations and proofs are written as pedagogically as possible. Numerical examples are also given in the appendices.

2.1 Derivatives payoffs

Hull defines derivatives as "a financial instrument whose value depends on (or derives from) the values of other, more basic, underlying variables" (Hull, 2008). In this subsection, we cover the payoffs of all the derivatives implemented in the web applications.

European options

The first class of derivatives implemented in the application are European call and put options. The former gives its holder the right, or opportunity, to buy the underlying asset at a predefined date for a predefined price. The latter is the same but the right given is to sell the underlying. The predefined date for European options is its maturity, i.e when it expires. The predefined price is called the strike or exercise price (Hull, 2008).

European options pay a cash flow $\psi(\cdot)$ at a certain time in the future, called the maturity T . At maturity, the payoff $\psi(T)$ (i.e., cash flow payment) is:

$$\psi(T) = (S_T - K)^+ = \max(0, S_T - K) \quad (1)$$

Where S_T is the underlying stock price at maturity and K the strike price. The "+" sign traduces the fact that the call is activated by the buyer only if the price of the share at time T (i.e. S_T) is higher than the strike price (K), and is mathematically translated by the maxima of either 0 and $S_T - K$.

By construction, the put option will simply be the opposite:

$$\psi(T) = (K - S_T)^+ = \max(0, K - S_T) \quad (2)$$

Note that in the case of a European options the payoff only depends on the events relative to time T , so European options are not path dependent (Hull, 2008).

Asian options

The payoff of Asian options depends on some type of average (arithmetic or geometric) of the underlying price during some period of the option's life (Hull, 2008). Their payoff depends thus on events relative to the whole path taken by the option, and are thus said to be path dependent. They are part of the exotic options family (Shreve, 2004a).

Defining $\bar{S}_T = \frac{1}{n} \sum_{i=1}^n S_{T_i}$ with $\{T_1 < \dots < T_n\}$ where T_i are the fixing dates and $T_n = T$ the option maturity, the payoff of the Asian call option is:

$$\psi(T) = (\bar{S}_T - K)^+ = \max(0, \bar{S}_T - K) \quad (3)$$

Note that Asian options are usually cheaper than European options, given their higher inherent risk caused by the path-dependency.

By construction, the put option is simply the opposite:

$$\psi(T) = (K - \bar{S}_T)^+ = \max(0, K - \bar{S}_T) \quad (4)$$

Exchange options

Also called Margrabe options, exchange options give the right to exchange one risky asset for another risky asset at maturity (Margrabe, 1978).

Consider at maturity giving up an asset 2, $S_T^{(2)}$, for an asset 1, $S_T^{(1)}$ (Vrins, 2020). The payoff of the exchange option is:

$$\psi(T) = (S_T^{(1)} - S_T^{(2)})^+ = \max(0, S_T^{(1)} - S_T^{(2)}) \quad (5)$$

2.2 Arbitrage-free pricing & risk-neutral valuation

The basic idea of pricing theory is that under some no-arbitrage assumptions, the unique and fair price is ensured if and only if the time t price of a derivative is given by the initial cost of a replication strategy (Bingham & Kiesel, 2004).

Indeed, by the law of one price, if such strategy exists, then the derivative price **must** be equal to the cost of the strategy.

The financial literature also shows that this price can also be derived through the risk-neutral expectation of the discounted derivative payoff.

This thesis' purpose is not to reinvent the wheel in terms of proofs or explanations, but rather to simply show that the replication strategy does perfectly replicate the pricing formula, i.e. showing its arbitrage-free nature, through the automation of all the lengthy computations. For this reason, all the explanations are kept short and pedagogical. Additionally, for the eventual curious reader, pedagogical and very detailed proofs and explications were written in the appendices.

Mathematically speaking, this is translated by:

$$\Pi_0 = V_0 = \mathbb{E}^{\mathbb{Q}} \left[\frac{\psi(T)}{\beta_T} \right] \quad (6)$$

Where Π_0 and V_0 are respectively the cost of the replicating strategy and the arbitrage-free derivative price at time 0 and β_T is the discounting factor at maturity. Note that the delta-hedging strategy Π_t will be further detailed later on.

In the following sub-subsections, the concepts of no-arbitrage price and risk-neutral valuation are shortly presented.

2.2.1 No-arbitrage price

Consider a trading strategy Π that perfectly tracks the cash flows of the derivative V , whatever happens to the underlying asset. Earlier, we stated that the replication strategy is the no-arbitrage price of the derivative. Let us show how that is the case by illustrating what happens if $\Pi_t \neq V_t$: one would inevitably be cheaper than the other and a smart investor could potentially simply buy one and sell the other.

This is shown as such: for $k > 0$, if the price of the derivative is $V_0 = \Pi_0 + k$, you can sell the derivative at $\Pi_0 + k$, launch the strategy (which only requires Π_0), and get a net profit of k today. At maturity, the strategy, given its replicating nature, will deliver exactly the derivative payoff (Vrins, 2020). Refer to **Table 4** for a visual.

$k > 0$	$t = 0$	$t = T$	$k < 0$	$t = 0$	$t = T$
buy strat.	$-\Pi_0$	$\psi(S_T)$	sell strat.	Π_0	$-\psi(S_T)$
sell deriv.	V_0	$-\psi(S_T)$	buy deriv.	$-V_0$	$\psi(S_T)$
arbitr. profit	k	0	arbitr. profit	k	0

Table 4: Arbitrage opportunity example

Thus, forcing $V_0 = \Pi_0$ is the only way to prevent arbitrage opportunities given the replicating nature of Π .

2.2.2 Risk-neutral valuation

In the previous sub-subsection, we showed that forcing $V_0 = \Pi_0$ is the only way to prevent arbitrage opportunities, and that Π is a strategy that replicates the cash flows of the derivative.

Let us now present the form that the derivative price V_0 takes: the risk-neutral price.

Typically, when students are asked how to price a derivative V , they will mention discounting the expected cash flows. The expectation probabilities and the discounting interest rate remain unclear (Vrins, 2020):

$$V_0 = \mathbb{E} \left[\frac{\psi(T)}{\beta_T} \right] \quad (7)$$

Indeed, what are the probabilities to use in the expectation? The probability of the underlying asset moving up or down? While this method of pricing is intuitive, it does not work in practice. Additionally, the risk preferences of the investor are also not being considered. Adjusting this formula for risk aversion levels requires computing a risk premium, which is very hard to do in practice (Gisiger, 2010).

Through the Fundamental Theorem of Asset Pricing (FTAP), modern pricing theory has shown that for replicable derivatives where the underlying does not pay dividends, **the unique price V_0 that prevents arbitrage opportunities (i.e., $V_0 = \Pi_0$) takes a mathematical form quite peculiar.**

First, let us introduce the concept of Martingale which will be required. The Martingale property states that the future expectation of a process is equal to its present value, given all the information about past events (Shreve, 2004b)(Vrins, 2020). In simple words, based on all the past information, the best estimate for the future value of the process is given by its present value (Shreve, 2004a).

With this knowledge, the FTAP shows that the unique price V_0 preventing arbitrage opportunities is:

$$V_0 = \mathbb{E}^{\mathbb{Q}} \left[\frac{\psi(T)}{\beta_T} \right] \quad (8)$$

Where the expectation probabilities are chosen such that the underlying discounted at the risk-free rate is a Martingale! This probability measure is called the risk-neutral measure.

A question often raised by students is the difference regarding the "normal" set of probabilities (called the historical $\mathbb{P}$ probability measure) and risk-neutral probabilities. The latter is bogus: it does not represent real-world probabilities! While in the real world investors will always demand a risk premium, FTAP shows under the $\mathbb{Q}$ -measure all assets have the same expected rate of return (the risk-free rate), and there is therefore no need to incorporate the investors' risk preferences (Bingham & Kiesel, 2004).

A more mathematical formal definition of risk-neutral pricing is given in [Appendix 8.1](#).

2.3 Standard pricing models

For both risk-neutral pricing and replication strategies, the underlying dynamics must be known.

In this subsection, we present the models that define these dynamics and give a framework for option pricing. We start in discrete time with the Cox-Ross-Rubinstein model and then in continuous time with the Black-Scholes-Merton model.

2.3.1 Cox-Ross-Rubinstein model

The Cox-Ross-Rubinstein model (CRR) is a binomial options pricing model, which was first initiated by William Sharpe in 1978 and then formalized by John Carrington Cox, Stephen Ross and Mark Rubinstein in 1979 (Cox, Ross, & Rubinstein, 1979).

The authors start with the following assumption: the stock dynamics are modeled by a multiplicative binomial process over discrete periods. Under the historical measure $\mathbb{P}$, the stock price S_i^j will either go up or down by known factors $u < 1$ and $0 < d < 1$ with positive probabilities p and $q = 1 - p$, and the log-returns are independent at all periods, i.e. from one period to another.

Observe in **Figure 3** a 1-period binomial tree.

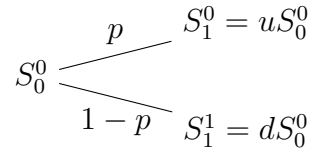


Figure 3: CRR: 1-period binomial tree

The model also assume constant interest rate, no transaction costs, and allowed short selling. The return of the interest rate r from one period to the following is $e^{r\delta}$ (continuous compounding, with $\delta = T/\text{tree periods}$). The drift of the stock is μ and its volatility σ . To rule out arbitrage opportunities, the following inequality must hold (Shreve, 2004a):

$$0 < d < e^{r\delta} < u \quad (9)$$

Additionally, the stock price at any period of the tree can be modeled as a function of a binomial random variable, and the constant factors u and d computed.

$$u = e^{\mu\delta + \sigma\sqrt{\delta}} \quad d = e^{\mu\delta - \sigma\sqrt{\delta}} \quad (10, 11)$$

It can also be said that the stock price follows a geometric random walk with drift $\mu\delta$ and volatility $\sigma\sqrt{\delta}$ (Vrins, 2020).

2.3.2 Black-Scholes-Merton model

Early in the 70s, Fischer Black, Myron Scholes and Robert Merton developed the Black-Scholes-Merton model. Historically, the model has been one of the most influential in quantitative finance. This success and influence were translated by an Nobel prize for economics in 1997 (Hull, 2008).

The model is based on the Black-Scholes-Merton partial derivative equation (BSM PDE). The price of all European-type derivatives are derived from this PDE.

$$\frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2} \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} = rV \quad (12)$$

Where V is the price of the option as a function of stock price S and time t , r is the risk-free interest rate, and σ is the volatility of the stock (Hull, 2008).

Its main assumptions are:

1. The underlying stock does not pay dividends, and its dynamics follow a geometric Brownian motion;
2. Transaction costs are non-existent;
3. Volatility and risk-free rate are assumed constant;
4. Fraction of shares can be traded.

Under the BSM framework, the underlying asset price is said to follow a geometric Brownian motion (GBM). Its stochastic differential equation (SDE) is given by:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \quad (13)$$

Where W is a Wiener process, also called a Brownian motion. It is characterized by the following properties:

1. It is grounded: $W_0 = 0$
2. It has independent increments: for any $s \leq t \leq q \leq r$, $(W_t - W_s) \perp\!\!\!\perp (W_r - W_q)$
3. The increments are stationary (Normal): $W_t - W_s \sim \mathcal{N}(0, t - s)$

2.4 Risk-neutral pricing formulas

In this subsection, we present the pricing formulas found through risk-neutral expectation for all the derivatives implemented in the web applications. We also give pedagogical numerical examples in the appendices.

Reminding the general form of the time- t pricing formula already given in equation (6),

$$V_t = \beta_t \mathbb{E}^{\mathbb{Q}} \left[\frac{\psi(T)}{\beta_T} | \mathcal{F}_t \right] \quad (14)$$

Cox-Ross-Rubinstein model

Under the CRR model in discrete time, the European and Asian options have been implemented in the web apps. In a 1-period CRR, the risk-neutral pricing formula can be derived to take the following form:

$$V_0^0 := e^{-r\delta} \left(V_1^0 \frac{e^{r\delta} - d}{u - d} + V_1^1 \frac{u - e^{r\delta}}{u - d} \right) \quad (15)$$

In a n -period setting, it is generalized to:

$$V_i^j = e^{-r\delta} \left(V_{i+1}^j \frac{e^{r\delta} - d}{u - d} + V_{i+1}^{j+1} \frac{u - e^{r\delta}}{u - d} \right) \quad (16)$$

Under the pricing measure, the price at all nodes is simply the discounted value of the two children nodes. The price tree is therefore filled backwards, starting from the leaves (i.e. the payoff) (Vrins, 2020). The proof to this formula is derived in the risk-neutral setting in [Appendix 8.3](#).

Simple numerical examples of CRR option pricing for both European and Asian options are given in [Appendix 8.4](#).

Black-Scholes-Merton model

Under the BSM model in continuous time, the European and Exchange options have been implemented in the web apps.

European options

Given a call option with payoff $(S_T - K)^+$, the risk-neutral expectation takes the following form:

$$\begin{aligned} V_t &= \beta_t \mathbb{E}^{\mathbb{Q}} \left[\frac{(S_T - K)^+}{\beta_T} | \mathcal{F}(t) \right] \\ &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [(S_T - K)^+ | \mathcal{F}(t)] \end{aligned} \quad (17)$$

Where $\beta_t = e^{rt}$ and r is the constant risk-free rate. The solution to this expectation is given by the well-known Black-Scholes pricing formula:

$$V_t = S_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2) \quad (18)$$

With $d_1 = \frac{\ln(S_t/K) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$, and $d_2 = d_1 - \sigma\sqrt{T-t}$. Note that the full proof is given for pedagogical reasons in [Appendix 8.6](#) and that the put option formula is simply derived through put-call parity.

Exchange options

Given the payoff $\psi(T) = (S_T^{(1)} - S_T^{(2)})^+$, the discounted risk-neutral expectation of the price is:

$$\begin{aligned} V_t &= \beta_t \mathbb{E}^{\mathbb{Q}} \left[\frac{(S_T^{(1)} - S_T^{(2)})^+}{\beta_T} | \mathcal{F}(t) \right] \\ &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [(S_T^{(1)} - S_T^{(2)})^+ | \mathcal{F}(t)] \end{aligned} \quad (19)$$

Solved in 1978 by Margrabe, the pricing formula is given by ([Margrabe, 1978](#)):

$$V_t = S_t^{(1)} \Phi(d_1) - S_t^{(2)} \Phi(d_2) \quad (20)$$

Where $d_1 = \frac{\ln(S_t^{(1)}/S_t^{(2)}) + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}$, $d_2 = d_1 - \sigma\sqrt{T-t}$ and $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}$.

ρ is the correlation between the two underlying stocks. Similarly to all previous formulas, the full steps to solve the expectation are given in [Appendix 8.9](#).

2.5 Replication strategy

As developed earlier, the time t price V_t of the derivative **must** be equal to the cost of the self-financing replicating strategy Π_t by the law of one price. Up until now, we have developed V_t from equation (6). In this subsection, we further develop the strategy Π_t and show under which assumptions it is perfectly equal to V_t .

For all pricing models, the logic of the strategy is always the same: the portfolio Π starts with a certain amount of money and is always composed of an equity and a cash account. The equity account holds at all times a certain amount of shares given by:

$$\Delta_t = \frac{\partial V_t}{\partial S_t} \quad (21)$$

And the remaining is left in cash. The portfolio equation is thus:

$$\Pi_t = \underbrace{\Delta_t S_t}_{\text{equity account}} + \underbrace{\Pi_t - \Delta_t S_t}_{\text{cash account}} \quad (22)$$

Finally, given the self-financing nature of the portfolio, the only changes in value are due to the stock appreciation/depreciation and the interests gained at the bank:

$$d\Pi_t = \Delta_t dS_t + r(\Pi_t - \Delta_t S_t)dt \quad (23)$$

The main idea behind the replicating strategy is that by holding Δ_t at all times and constantly re-balancing the portfolio by selling and buying the underlying stock, the strategy **hedges away** the randomness embedded in the underlying stock. **This allows the strategy Π_t to perfectly replicate the option price V_t whatever the stock trajectory (i.e., $\Pi_t = V_t$)!** For this reason, it is named the delta-hedge strategy ([Vrins, 2020](#)).

Cox-Ross-Rubinstein model

Under a n-period CRR model, the re-balancing portfolio equation (23) is given by:

$$\Pi_i^j = \Delta_i^j S_i^j + e^{r\delta}(\Pi_i^j - \Delta_i^j S_i^j) \quad (24)$$

Then, for both European and Asian options, Δ_i^j is the same. Most importantly, the continuous derivative $\Delta_t = \frac{\partial V_t}{\partial S_t}$ becomes a discrete derivative, and thus a difference quotient for the option and stock prices. It is computed as such:

$$\Delta_i^j = \frac{V_{i+1}^j - V_{i+1}^{j+1}}{S_i^j(u - d)} \quad (25)$$

With this Δ_i^j value, the portfolio perfectly replicates the risk-neutral option price shown in equation (16). However, the computation of such strategies is lengthy and tedious. A simple 2-period tree is given for pedagogical purposes, as illustrated in [Appendix 8.5](#), for both European and Asian options.

This highlights the need for a tool that automates the computations such that it can be checked easily.

Note that the derivation of Δ_i^j is the same for both European and Asian options and is derived through replicating portfolio in [Appendix 8.2](#).

Black-Scholes-Merton model

In the continuous time model, the derivation of Δ_t follows from the option pricing formulas showed earlier.

European options

From the call option price $V_t = S_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2)$, one can find that:

$$\begin{aligned} \Delta_t &= \frac{d(S_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2))}{dS_t} \\ &= \Phi(d_1) \end{aligned} \quad (26)$$

Shortly put, as long as one the strategy holds $\nu \Phi(\nu d_1)$ ($\nu = 1$ for calls and -1 for puts) at all times and re-balance the portfolio accordingly, it can perfectly replicate the BSM pricing formula. The derivation of Δ_t can be found in [Appendix 8.7](#). The mathematical development explaining **why** the delta-hedging strategy works is given in [Appendix 8.8](#).

A visual of the strategy is given in **Figure 4**, where δt is the strategy re-balancing step.

t	cash	equity	portfolio
0	Π_0	—	Π_0
	$\Pi_0 - \Delta_0 S_0$	$\Delta_0 S_0$	Π_0
δt	$(\Pi_0 - \Delta_0 S_0)(1+r\delta t)$	$\Delta_0 S_{\delta t}$	$\Pi_{\delta t}$
	$\Pi_{\delta t} - \Delta_{\delta t} S_{\delta t}$	$\Delta_{\delta t} S_{\delta t}$	$\Pi_{\delta t}$
...	...	...	...
...	...	...	...
$(k-1)\delta t$	$(\Pi_{(k-1)\delta t} - \Delta_{(k-1)\delta t} S_{(k-1)\delta t})$	$\Delta_{(k-1)\delta t} S_{(k-1)\delta t}$	$\Pi_{(k-1)\delta t}$
$k\delta t$	$(\Pi_{(k-1)\delta t} - \Delta_{(k-1)\delta t} S_{(k-1)\delta t})(1+r\delta t)$	$\Delta_{(k-1)\delta t} S_{k\delta t}$	$\Pi_{k\delta t}$
	$\Pi_{k\delta t} - \Delta_{k\delta t} S_{k\delta t}$	$\Delta_{k\delta t} S_{k\delta t}$	$\Pi_{k\delta t}$
...	...	...	...

Figure 4: Continuous time European option replicating strategy

Source: taken from (Vrins, 2020)

The formulas showcased in the visual seem very complicated and abstract. It seems difficult to implement in practice. Once gain, this showcases the utility of a tool that makes the computation of the replication strategy automatic, as students often need "to see that it works" to believe in it.

Exchange options

From the option price $V_t = S_t^{(1)}\Phi(d_1) - S_t^{(2)}\Phi(d_2)$, one can find that:

$$\Delta_t^{(1)} = \Phi(d_1) \quad \Delta_t^{(2)} = -\Phi(d_2) \quad (27, 28)$$

The mathematical development explaining **why** the delta-hedging strategy works is given in [Appendix 8.10](#).

It is also interesting to note that for exchange options, the portfolio re-balancing equation can be simplified. Indeed, for the replication of exchange options, the cash is in reality always perfectly allocated between both stocks. There is thus no cash in the cash account.

The re-balancing equation of the portfolio is therefore simplified into:

$$d\Pi_t = \Delta_t^{(1)} dS_t^{(1)} + \Delta_t^{(2)} dS_t^{(2)} \quad (29)$$

3 Development and deployment of a web-based application

In this section, the complete methodology behind the tool is presented. It is structured in three subsections: the app development framework, a step-by-step tutorial on how to create and deploy an app, and the legal aspects of the tool. The goal is thus to ensure the extendibility goal of this thesis.

Note that the provided methodology is the one **applied for this specific kind of web application** and is not unique. Other methodologies obviously exist. It is the methodology that the author implemented and this section's purpose is to simplify the life of the future students that will further extend this work.

3.1 App development framework

This subsection covers the framework used in the development of the app: the **Dash library**, the **Heroku cloud platform** and **Git version control system**.

3.1.1 Dash library

Dash is a Python library that is open-source. It is used for the creation of interactive web-based applications ([Plotly, 2017](#)).

Typically, web development is based on multiple programming languages: JavaScript, HTML and CSS. The Dash library is very useful given that it allows the development of modern-looking web applications in pure Python language. The library takes care of all the JavaScript, HTML and CSS behind the scenes.

Dash applications are composed of two parts: the layout and the interactions. The former defines the look of the app, and the latter is self-explanatory ([Mwiti, 2018](#)).

By default, Dash apps run on localhost, i.e. the user's computer. For other users to have access, the Dash app has to be hosted in an online server.

3.1.2 Heroku cloud platform

Heroku is a container-based cloud Platform as a Service (PaaS) supporting numerous programming languages. Simply put, it hosts web applications on the internet ([Heroku, n.d.](#)).

In our work, the Heroku cloud platform is used to host the Dash application, and allows anyone with the URL link to have access to the app.

The hosting service provided by Heroku is free of charge for applications incurring low online traffic, i.e. under 550 hours of usage through all applications per Heroku account ([Heroku, 2021](#)).

An interesting service of Heroku is its customization of the app domain at no extra charges. By default, all Heroku applications are under the form [name of app].herokuapp.com. With the customization, the application could be under www.[name of app].com ([Heroku, 2020a](#)). This could prove to be useful if the apps are to be used under the university web domain.

3.1.3 Git version control system

A version control is a system that records changes to software over time such that the programmer can recall specific versions at a later date ([Chacon & Straub, 2014](#)).

Such systems are mostly used in software engineering to properly manage programs, and keep track of all changes. It is part of software configuration management ([O’Sullivan, 2009](#)).

The version control system required by Heroku is Git. It is used for the deployment and management of the hosted Dash applications. Its purpose is to be the link between the localhost and the Heroku cloud platform host.

3.1.4 Workflow

This subsection summarizes visually the earlier framework through **Figure 5**.

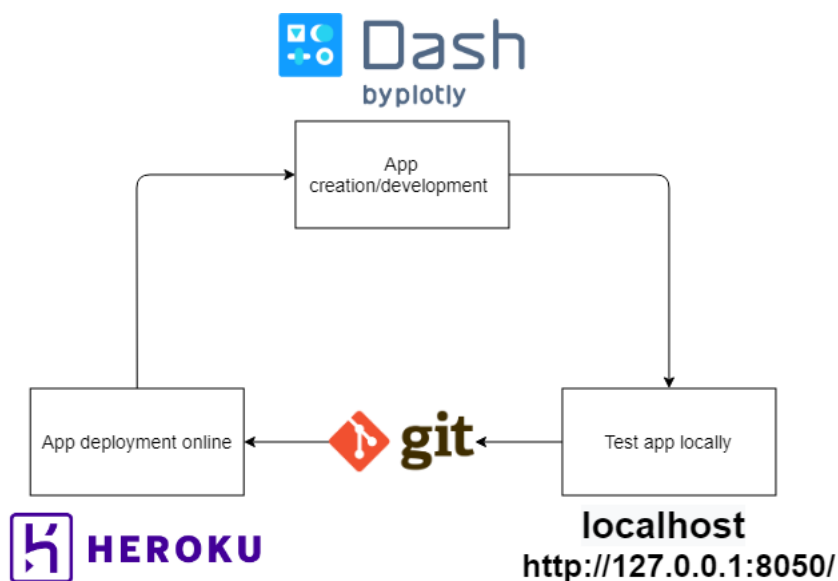


Figure 5: App development workflow

In summary, the Python Dash library is used to create the app and design it. It only runs locally and is used for testing purposes. Then, using Git as a version control system, the app is deployed online and hosted in the Heroku servers.

3.2 Create & Deploy your own app: step by step

This subsection's purpose is to provide a simple step-by-step methodology for the creation and deployment of similar Dash applications.

As explained in [section 1 Problem specifications and design choices](#), one of this thesis' goals is to be extendable, i.e. that other derivatives' replication strategies could be implemented in a similar manner. For this endeavor, such a step by step methodology could prove to be useful for future students that will extend this thesis.

This tutorial is for the **creation and deployment** of the application, and does not intend to teach the Dash library. There already are numerous resources and documentation for that online.

Note that while this methodology is written for the operating system Windows 10, all commands are valid for macOS or Linux. Additionally, it takes as basis Rankin’s work and is further polished (Rankin, 2019).

3.2.1 Software requirements

In this sub-subsection, an exhaustive list of all required software is given, along a download link. These software **must** be downloaded and installed. See in **Table 5**.

	Description	Official URL
Python	Object-oriented programming language	https://www.python.org/downloads/
Dash library	Python package	https://dash.plotly.com/installation
Heroku	Cloud platform	https://devcenter.heroku.com/articles/getting-started-with-python
Git	Version control system	https://git-scm.com/download/win

Table 5: All software required to create your own application

We remind that these software are for the creation of **similar apps** to this thesis. Other software and methodologies naturally exists.

Additionally, it is highly recommended to use a text editor for code, such as Sublime Text, or for most advanced users a development environment such as Visual Studio Code or PyCharm.

Note that other Python libraries will be required depending on the app’s functionalities.

3.2.2 Configuration

Once installed, both Git and Heroku need to be configured. The softwares are used through commands, i.e lines of code. The commands are inputted in the operating system command-line interpreter. For windows, it is the Command Prompt, referred as CMD.

By default, running CMD is done by either searching for it on Windows’ search bar, or using the **Win + R** shortcut and writing CMD.

Setting-up Git

Run these two Git commands to set-up your Git identity in CMD. This initial step's purpose is crucial: for any future modifications of the application code, i.e. commits, Git will recognize it as the user's, and timestamp it. Refer to **Figure 6**.

```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\WindowsAccount>git config --global user.name "FirstName LastName"

C:\Users\WindowsAccount>git config --global user.email FirstNameLastName@example.com
```

Figure 6: Git identity set-up CMD commands

More information regarding the configuration of Git can be found on this link: [First-Time Git Setup](#), which is also given in (Chacon & Straub, 2014).

Setting-up Heroku

Similarly to Git, it is required to define your identity on Heroku. It is done by creating an Heroku account: [Heroku - Sign up](#). Once that is done, it is required to log in to the newly created Heroku account from CMD. It will allow the user to create, deploy and modify applications directly from CMD.

To do so, run the `heroku login` command as shown in **Figure 7**:

```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\WindowsAccount>heroku login

heroku: Press any key to open up the browser to login or q to exit
- Warning: If browser does not open, visit
- https://cli-auth.heroku.com/auth/browser/**
heroku: Waiting for login...
Logging in... done
Logged in as me@example.com
```

Figure 7: Heroku login CMD command

The `heroku login` command opens on your default web browser the Heroku login page. "This authentication is required for both the `heroku` and `git` commands to work correctly" (Heroku, 2020b).

3.2.3 Creating the app

At its core, a Dash application is simply a Python script where the Dash library is imported, and the app's structure defined using Python's code.

As mentioned above, this tutorial is for the creation and deployment of a Dash application, not teaching the Dash library. The best documentation to write a Dash application is by far the official resources. The documentation is extensive, and starts from the very basics of Dash: [Starting Dash](#).

Naturally, the thesis' applications are open-source and can be used as a basis. They can be found in the author's GitHub: github.com/MichelVanderhulst

3.2.4 Deploying the app

This sub-subsection covers the different steps required in the deployment of the app.

The steps are and must be done in the following order:

1. Setting-up a new repository with Git
2. Creating the Heroku app
3. Deploying the app online

Firstly, for the Git version control system to work, it needs a virtual storage of the files. The virtual storage is the repository (repo) ([Chacon & Straub, 2014](#)). It needs to be initiated in the application's source folder. That is done using the `git init` command.

Then, some files are required for Heroku. These files can be downloaded from the author's [GitHub repository](#), and must be located in the app's source folder. After that, the following Git commands can be used `git add .` and `git commit -m"app creation"`.

The command `git add .` promotes all changes done in the application's source folder. Then, `git commit -m"some comment"` records the current version of the folder, changes from the prior command included ([Chacon & Straub, 2014](#)).

Finally, to push the application to the Heroku online servers, this last command must be used: `git push heroku master`. Note that this assumes no errors are in the Python code, which can be checked beforehand locally.

3.2.5 Maintaining and extending the app

Applications often require updates, improvements and mistakes handling. In the current framework, it is equal to re-deploying the application.

With Heroku, once the app is online, it cannot be modified unless it is re-deployed. Updating the app is done by editing the source Python files, and re-deploying the application using the earlier commands `git add .`, `git commit -m"some comment relating to the update"` for the commit, and finally `git push heroku master` to re-deploy it.

See **Figure 8** as example of updating an application. In this case, the update was regarding a change in a plot.

```
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\WindowsAccount\YourAppFolder>git add .
C:\Users\WindowsAccount\YourAppFolder>git commit -m"changes in greeks plot"
[master cf33e56] changes in plots
 2 files changed, 37 insertions(+), 204 deletions(-)
 rewrite app.py (23%)
```

Figure 8: Updating application example

In terms of **extensions** beyond this work, e.g. new app functionalities, future students will simply have to access the Heroku accounts of the currently implemented applications, which will naturally be shared with this thesis supervisor, Prof. Vrans.

Using the Heroku accounts, future students will be able to edit the code of the existing applications and create their own.

3.3 App legal aspects

This subsection covers the legal aspects of the tool. Indeed, given the online nature of the tool, along its future usage, legal aspects of all its components had to be thoroughly checked.

The first component is the code of the app itself. Wishing for it to be open-source, the app is released under a permissive MIT license. Under this license, the app has close to limitations on its utilisation (*The MIT License*, n.d.). An example of the MIT license used can be found in [Appendix 8.11](#).

Note that all the Python libraries used were specifically chosen such that they also operate under the MIT license.

The second legal component is the Heroku cloud platform. Under its current terms and conditions, it holds no legal rights on the hosted applications. Its only purpose is to host them on its web servers.

4 Presentation of created web applications

This section presents the solution to the addressed problem of this project thesis, i.e. the derivatives replication strategy web applications.

As a reminder, the following web applications were created:

- In discrete time with the CRR model: European and Asian options
- In continuous time with the BSM model: European and Margrabe options

For pedagogical and consistency purposes, all applications follow the same code architecture and present similar functionalities.

4.1 App code architecture

We start by showcasing the apps' code architecture. This refers to the structure and organization of the app's code (as seen in the top quadrant of **Figure 5**). Find in **Figure 9** the architecture of all the currently implemented web applications.

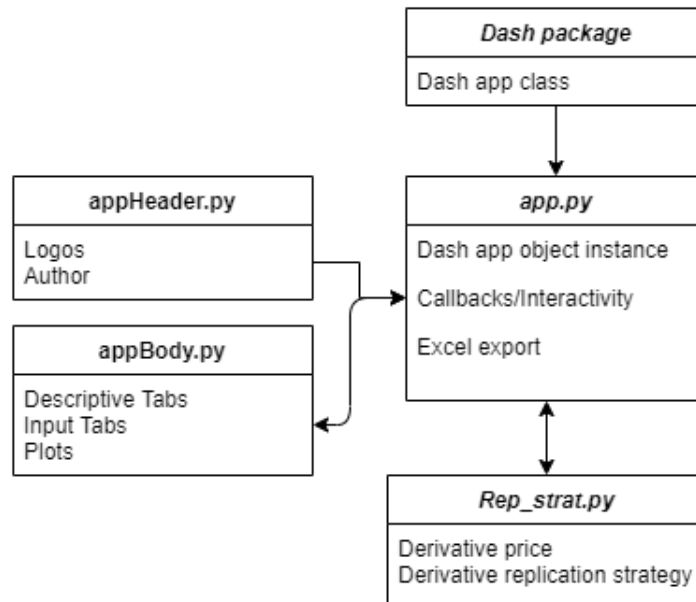


Figure 9: App code architecture

By construction, it is object-oriented programming as all apps are objects, i.e. instances, of the app class of the Dash package. As highlighted in [section 1 Problem specifications and design choices](#), its benefits are **clear design structure, re-usability of software and ease of code maintenance**.

This is shown here: the structure of the code is quite simple, given that only an instance of the Dash app class has to be created. Its already-existing methods are then used to design its structure as simply a header and a body.

The interactivity of the app is ensured as such: the app boots up with default input values, which are given to the replication strategy function created during this work. Plots are generated, and sent to the app body. Then, every time the user changes any input in the app body, the app reacts to it by calling dynamically the *Rep_strat.py* script to once gain generate the visuals.

This simple flow ensures ease of code maintenance and re-usability of the apps. Most importantly, it is only possible due to our choice of object-oriented programming.

4.2 Apps functionalities

This subsection's purpose will be to showcase the apps created for this thesis. Most notably, emphasis will be put on the:

- **Functionality:** how well the apps serve their pedagogical objective.
- **Model limitations:** how the apps can be used to highlight some of the models limitations.

Given that all the apps are quite similar, one example for each model will be presented: the European option under the Cox-Ross-Rubinstein model and the Margrabe option under the Black-Scholes-Merton model.

Note that the general content of the major app sections was already defined in [section 1 Problem specifications and design choices](#). As a reminder, these were: presentation of tool and asset, mathematical model used to price asset, methodology of implementation and replication strategy construction, inputs and plots.

4.2.1 Example 1: European option CRR web app

The app can be found at the following URL: eu-option-crr.herokuapp.com. Refer to **Figure 10** for the starting screen users are welcomed to.

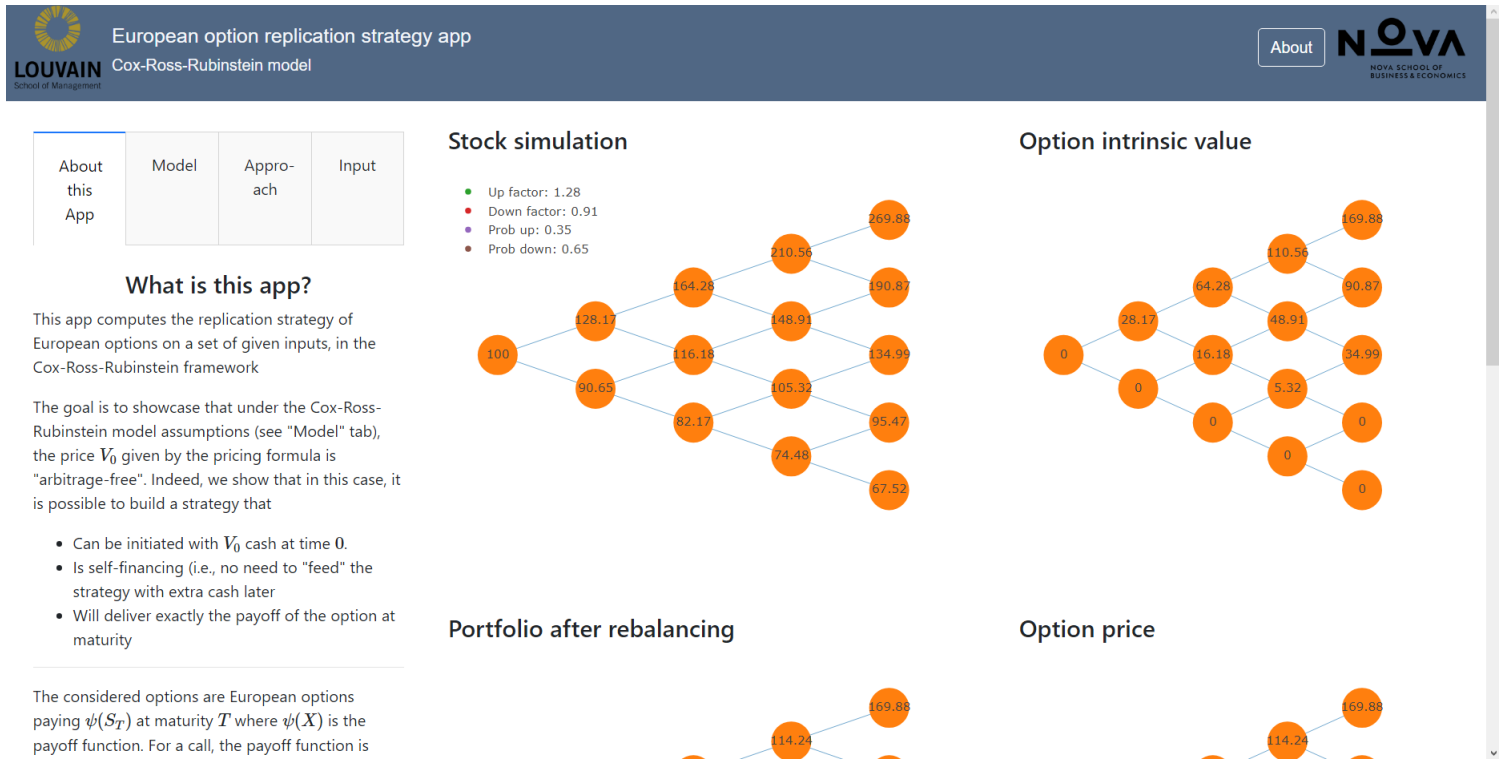


Figure 10: Starting screen of European option (CRR) app

Sub-tabs

On the left-hand side of the app, there is a main tab with multiples sub-tabs. The user is welcomed by the "About this App" sub-tab, which presents the app, its pedagogical goal and the type of derivative.

Then comes the "Model" sub-tab, which explains model assumptions, the underlying asset dynamics and the risk-neutral pricing of the option. The "Approach" sub-tab follows with the presentation of the self-financing replication strategy. All explanations are given in a pedagogical style. Note that all math equations are displayed in $\text{\LaTeX}$ style for aesthetic reasons.

Finally comes the most important part, the **"Input"** sub-tab. It deserves its own illustration, and can be seen in **Figure 11**.

Hover your mouse over any input to get its definition.

European Call option

Spot price 100

Strike 100

Drift: 10%

Volatility: 20%

Risk-free rate: 5%

Maturity: 3 years

Tree periods: 4

Graph type: ☐ Spatial ☒ Tree

Note that some errors are possible due to rounding decimals when displaying the values in the chart. Download the exact values in excel format if you wish to check.

Download .xlsx

Figure 11: Input sub-tab of the European option (CRR) web app

Plots

In the right-hand side of the app, six plots are given: the stock simulation, the option intrinsic value, the replicating portfolio, the option price, the number of held Δ shares held and the cash account.

In the first binomial tree of the stock, the user is given the CRR up and down factors along the risk-neutral probabilities computed from his chosen input values. Below that, the user can find the replicating portfolio and risk-neutral option price trees. The user can there see how perfectly equal these trees are! Most importantly, the user can easily check for himself this equality by looking at the Δ and cash account trees, whatever his inputs are.

A lot of emphasis was put on giving the user the utmost freedom in terms of input customization.

The most notable app functionalities in the CRR model are:

- Lot of inputs can be used to play around the option price and replication strategy. This is useful for sensitivity analysis.
- User input validation (only positive values for the spot and strike, ...) for robustness purposes. A example is given in [Appendix 8.14](#).
- Changing the graph type from a binomial tree to a spatial (x,y) coordinate plot.
- Excel export of the generated data. A example is given in [Appendix 8.12](#)

4.2.2 Example 2: Exchange option BSM web app

The app can be found at the following URL: exchange-option-bsm.herokuapp.com. Refer to Figure 12.

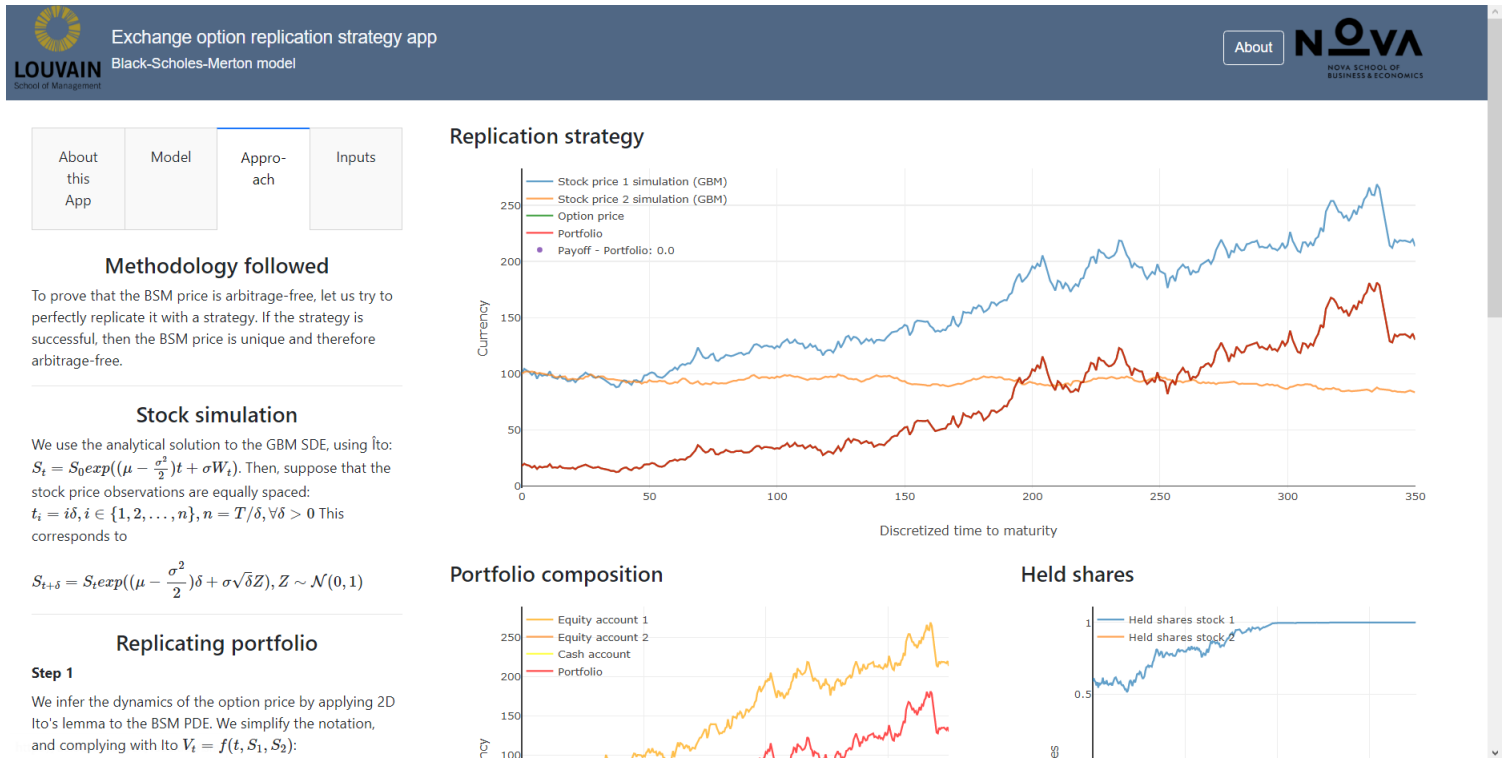


Figure 12: Exchange option (BSM) app

Sub-tabs

Here, the "Approach" sub-tab and the plots of the app are shown. As mentioned earlier, a lot of emphasis and effort was put into the pedagogy of the explanations. The goal is for students to understand the concept of self-financing replicating strategies, i.e. graphs alone are not enough.

The **”Inputs”** sub-tab is similar to the previous model but present some functionalities that add a lot of pedagogical value:

Place your mouse over any input to get its definition.

Spot price 1

Spot price 2

Stocks correlation: 0.3

Drift stock 1: 10%

Volatility stock 1: 25%

Drift stock 2: -10%

Volatility stock 2: 10%

Risk-free rate: 3%

Maturity: 3.5 years

Discretization step (dt)

Time between two rebalancing (in dt unit)

Transaction costs

☒ No TC ☐ Fixed TC ☐ Proportional TC

Change stock trajectory

The current stock trajectory scenario is:

[Download Data](#)

Figure 13: Input sub-tab of the Margrabe option (BSM) web app

Plots

In the right-hand side of the app, three plots are given for the Margrabe app: the replication strategy, the portfolio composition and the held shares Δ .

In the first plot, the user can find the two stock trajectories, the option price computed with the Margrabe formula and the replicating portfolio. The user can see that whatever his inputs, the two curves are always the same: the replicating portfolio does his job!

The "quality" of the replicating strategy is also shown in the legend, given by the difference of option payoff and portfolio value.

The same freedom in terms of input customization was given for the continuous time model along the excel exportation and input data validation for robustness purposes. Note that an example of the generated excel file is given in [Appendix 8.13](#).

More interestingly, the frequency of portfolio re-balancing and transaction costs are also inputs.

These limitations of the BSM model are often very difficult to consider analytically. Automating their computations in the replication strategy context therefore adds a lot of pedagogical value to the applications.

Finally, given that the stocks follow geometric Brownian motions which are randomly generated, more than 500,000 stock trajectories were computed, which the user can choose from as he wishes. This gives a lot of sensitivity analysis freedom.

The portfolio composition plot gives the equity and cash accounts. In this manner, the user can visualize how shares are bought and sold to perfectly delta-hedge the option.

A notable feature of the generated plots is their high interactivity: placing the mouse over the lines will dynamically display their values, the user can freely zoom in and out, download the plot as an image in .png format, and even hide curves by clicking on them if he wishes to focus his attention on some of them.

5 Future work

This section presents the future improvements and extensions that could be implemented in the application.

5.1 Design

The current design of the apps was implemented with pedagogy and practicability in mind. As such, its design could use some improvements.

5.1.1 Application appearance

As presented in section 3.1.1, the Dash library allows pure Python code to develop web applications. It is done through working the required JavaScript and HTML behind the scenes.

Dash applications can be fed with external style sheets, written in native CSS code. In this manner, the applications' appearance could definitely be improved.

5.1.2 Design responsiveness

Responsive design is defined as the creation of websites such as to provide an optimal viewing and interaction experience depending on the device and browser characteristics (Sarabadani Tafreshi, Marbach, & Norrie, 2017).

Currently, the applications' responsiveness to the user's device or browser is quite limited. The user experience suffers when accessing the apps with smartphones or with computers with very small screens.

Future improvements in this direction could be done with external JavaScript style sheets.

5.2 Performance

The two main goals of this work were its pedagogy and extendibility. Performance was not considered that important given that the web applications will not be used for commercial purposes.

Nevertheless, it is a nice-to-have feature. Indeed, when pushing the input to their extremes, the apps' speed suffers. For example, this is observed when increasing the number of tree

periods for the CRR applications, or by decreasing the discretization step for the BSM applications.

This limitation is mainly caused by the Python language itself. Python code, and most precisely Python loops, run more slowly than in Java or C++ ([Shein, 2015](#)).

A potential solution could be the usage of vectorization, but this would come at the cost of code readability. Another candidate is the usage of packages such as Cython or Numba, which would once again cost code readability, but significantly less than vectorization. We believe this last candidate to be a very good idea in the long-term.

5.3 Extensions

Extensions are a natural part of applications. This is even more true for this work given its pedagogical goal. The list of replicable derivatives, i.e. of potential candidates, is numerous. Among them are digital, quanto and maximum options.

For pedagogy's sake, we believe that they could use similar applications.

6 Conclusion

To conclude this work, let us first restate this thesis' purpose. When teaching quantitative finance, **explaining to students that initial cost of a self-financing replicating strategy corresponds to the price of a derivative is difficult**. Students have trouble visualizing it and hand calculation is both unpractical and very limiting.

We tackled this problem by **developing open-source web applications that automate the computations** for a set of user-chosen input, and display the corresponding trading strategies. In this manner, students can visualize the strategy that is equal to the derivative price at all times, thus helping polish their financial intuition and knowledge. In simple words, students can see for themselves that the strategy does work.

The **two main objectives pursued in this thesis were its pedagogical vocation and its extendibility**. These goals were ensured by defining the tool's desired characteristics: usability, robustness, deployability, maintainability and extendibility. Then, the characteristics were implemented through various design choices. For example, the robustness characteristic was taken care of by following software engineering standard practices and extensive error handling of problematic inputs. Deployability and extendibility were implemented by using the Python programming language, which research showed to be the most promising and of high usage in the financial industry. That way, both the author and future students would not have to worry about legacy code or developing coding skills in a language not useful for their career.

In terms of future works, we sincerely hope for future Master's thesis to further extend this work by adding new replicable derivatives and functionalities. **For all stakeholders, i.e. students and universities, we believe that having a great library of derivatives pricer/hedger adds a lot of value to the Finance programs.**

Finally, as in all research processes, problems were raised at the same time as others were solved, especially regarding the application's implementation. We highlighted the relevant points of future improvements, such as the application's design responsiveness or appearance.

7 References

- Azizieh, C. (2017). *Applications of black-scholes model*. [PowerPoint slides]. Retrieved from Vrije Universiteit Brussel.
- Bingham, N., & Kiesel, R. (2004). *Risk-neutral valuation: Pricing and hedging of financial derivatives* (2nd ed.). Springer-Verlag London.
- Bodart, A., Henry, B., Kinart, A., & Vanderveken, R. (2018). *Pricing exchange options*. Louvain School of Management: unpublished group assignment.
- Chacon, S., & Straub, B. (2014). *Pro git* (2nd ed.). Apress.
- Cox, J. C., Ross, S. A., & Rubinstein, M. (1979). Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3), 229 - 263. doi: [https://doi.org/10.1016/0304-405X\(79\)90015-1](https://doi.org/10.1016/0304-405X(79)90015-1)
- Eça, A. (2020). *Lecture 2: Principles of model building*. [PowerPoint slides]. Retrieved from Nova School of Business and Economics 2270 Financial Modeling.
- Fernandez, J.-C., Mounier, L., & Pachon, C. (2005). A model-based approach for robustness testing. In *Ifip international conference on testing of communicating systems* (pp. 333–348).
- Gisiger, N. (2010). Risk-neutral probabilities explained. *Capital Markets: Asset Pricing Valuation*.
- Harrison, J., & Kreps, D. M. (1979). Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, 20(3), 381-408. Retrieved from <https://www.sciencedirect.com/science/article/pii/0022053179900437> doi: [https://doi.org/10.1016/0022-0531\(79\)90043-7](https://doi.org/10.1016/0022-0531(79)90043-7)
- Harrison, J., & Pliska, S. R. (1981). Martingales and stochastic integrals in the theory of continuous trading. *Stochastic Processes and their Applications*, 11(3), 215-260. Retrieved from <https://www.sciencedirect.com/science/article/pii/0304414981900260> doi: [https://doi.org/10.1016/0304-4149\(81\)90026-0](https://doi.org/10.1016/0304-4149(81)90026-0)
- Heroku. (n.d.). *What is heroku*. Retrieved from <https://www.heroku.com/what> (Accessed on August 8, 2020)
- Heroku. (2020a). *Custom domain names for apps: Heroku dev center*. Retrieved from <https://devcenter.heroku.com/articles/custom-domains> (Accessed on August 13, 2020)
- Heroku. (2020b). *Heroku - getting started with python*. Retrieved from <https://devcenter.heroku.com/articles/getting-started-with-python?singlepage=true> (Accessed on August 30, 2020)
- Heroku. (2021). *Heroku pricing: Flexible pricing options, from personal projects to enterprise*

- applications, and everything in between*. Retrieved from <https://www.heroku.com/pricing> (Accessed on February 8, 2021)
- Hilpisch, Y. (2015). *Derivatives analytics with python: Data analysis, models, simulation, calibration and hedging* (1st ed.). Wiley.
- Hull, J. (2008). *Options, futures and other derivatives* (7th ed.). Prentice Hall.
- Lei, K., Ma, Y.-N., & Tan, Z. (2014). Performance comparison and evaluation of web development technologies in php, python, and node.js. *2014 IEEE 17th International Conference on Computational Science and Engineering*, 661-668.
- Margrabe, W. (1978). The value of an option to exchange one asset for another. *Journal of Finance*, 33(1), 177-86.
- The mit license*. (n.d.). Retrieved from <https://opensource.org/licenses/mit-license.html> (Accessed on August 8, 2020)
- Morley, D., & Parker, C. S. (2016). *Understanding computers: Today and tomorrow comprehensive* (16th ed.). Course Technology Press.
- Murray, S. (2019, Jun 17). Graduates with tech and finance skills in high demand. *The Financial Times*. Retrieved from <https://www.ft.com/content/6c7001ec-70e3-11e9-bf5c-6eeb837566c5> (Accessed on March 28, 2020)
- Mwiti, D. (2018, Aug). *Dash for beginners*. Retrieved from <https://www.datacamp.com/community/tutorials/learn-build-dash-python> (Accessed on August 8, 2020)
- Mössenböck, H. (1995). *Object-oriented programming in oberon-2* (2nd ed.). Springer-Verlag Berlin Heidelberg.
- O'Sullivan, B. (2009). *Mercurial: The definitive guide* (1st ed.). O'Reilly Media Inc.
- Plotly. (2017, Nov). *Introducing dash*. Author. Retrieved from <https://medium.com/plotly/introducing-dash-5ecf7191b503> (Accessed on August 8, 2020)
- Poulsen, R. (2010). The margrabe formula. *Encyclopedia of Quantitative Finance*, 1118-1120.
- Python. (2020). *Python webframeworks*. Retrieved from <https://wiki.python.org/moin/WebFrameworks> (Accessed on March 28, 2020)
- Rankin, Z. (2019, Feb 20). *A gentle introduction to dash development and deployment*. Towards Data Science. Retrieved from <https://towardsdatascience.com/a-gentle-introduction-to-dash-development-and-deployment-f8b91990d3bd> (Accessed on August 19, 2020)
- Richards, M., & Ford, N. (2020). *Fundamentals of software architecture: An engineering approach*. O'Reilly Media. Retrieved from <https://books.google.be/books?id=xa7MDwAAQBAJ>
- Sarabadani Tafreshi, A. E., Marbach, K., & Norrie, M. C. (2017). Proximity-based adap-

- tation of web content on public displays. In J. Cabot, R. De Virgilio, & R. Torlone (Eds.), *Web engineering* (pp. 282–301). Cham: Springer International Publishing.
- Shein, E. (2015). Python for beginners. *Communications of the ACM*, 58(3), 19–21.
- Shreve, S. E. (2004a). *Stochastic calculus for finance ii continuous-time models* (2nd ed.). Springer Finance.
- Shreve, S. E. (2004b). *Stochastic calculus for finance i the binomial asset pricing model* (2nd ed.). Springer Finance.
- Thakur, D. (2019). *What is software? characteristics and classification of software*. Retrieved from <https://ecomputernotes.com/software-engineering/characteristics-and-classification-of-software> (Accessed on May 17, 2021)
- TIOBE. (2020a). *About tiobe company*. Retrieved from <https://www.tiobe.com/company/about/> (Accessed on March 25, 2020)
- TIOBE. (2020b). *Tiobe index for march 2020*. Retrieved from <https://www.tiobe.com/tiobe-index> (Accessed on March 25, 2020)
- Vanderhulst, M., Saaïdi, A., & VanParijs, S. (2019). *Pricing barrier options*. Louvain School of Management: unpublished group assignment.
- Vrins, F. (2020). *Derivatives pricing*. [PowerPoint slides]. Retrieved from Louvain School of Management LLSM2225 Derivatives Pricing.
- Yan, Y. (2017). *Python for finance ed. 2* (2nd ed.). Packtpub.
- Yu, X., & Xie, X. (2013). On Derivations of Black-Scholes Greek Letters. *Research Journal of Finance and Accounting*, 4(6), 200–219. (ISSN 2222-1697 (Paper) ISSN 2222-2847 (Online))

8 Appendices

8.1 Formal definition of risk-neutral pricing

After intuitively defining the concept, we give a more mathematical in-depth presentation of risk-neutral valuation.

We start by shortly explaining the **concept of martingale**: relative to a filtration $\mathbb{F}$, a process X is said to be a martingale if and only if ([Vrins, 2020](#)):

$$\mathbb{E}[X_t | \mathcal{F}(s)] = X_s, \forall s \leq t \quad (30)$$

Based on the information at time s , the best estimate for the process X_t at time t for $s \leq t$ is given by X_s ([Shreve, 2004a](#)).

Introduced by Dybvig and Ross in 1987, the **Fundamental Theorem of Asset Pricing (FTAP)** is a term used "for theorems establishing the equivalence of an economic modeling condition such as no-arbitrage to the existence of the mathematical modeling condition existence of equivalent martingale measures" ([Bingham & Kiesel, 2004](#)).

At its core, it combines two theorems (cited from ([Bingham & Kiesel, 2004](#))):

- No-arbitrage theorem: "a market $\mathcal{M}$ is arbitrage-free if and only if there exists a probability measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ under which the discounted asset price process is a $\mathbb{Q}$ -martingale."
- Completeness theorem: "an arbitrage-free market $\mathcal{M}$ is complete if and only if there exists a **unique** probability measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ under which discounted asset prices are martingales."

Therefore, FTAP states that in an arbitrage-free complete market $\mathcal{M}$, there exists a unique equivalent martingale measure $\mathbb{Q}$.

The risk-neutral pricing formula takes thus seen in equation (6).

In the following appendix, we derive the CRR pricing formula with a replicating portfolio. We later show how it can be derived using the Martingale property, thus illustrating the FTAP in action.

[\[Return to 2.2.2 Risk-neutral valuation\]](#)

8.2 CRR pricing formula through replicating portfolio

Defining the stock S_i^j , cash account B_i^j a derivative V_i^j paying $\psi(T)$ in a 1-period CRR model in Figure 14. Assume that $T = 1$, and $\delta = 1$ for simplicity.

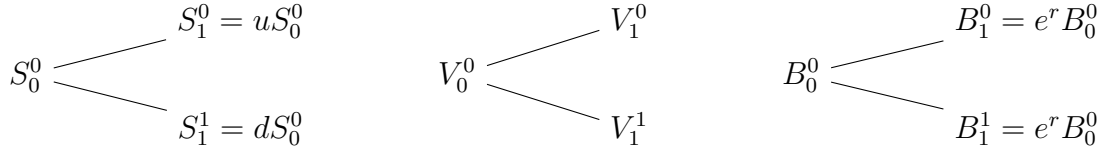


Figure 14: CRR: 1-period tree for stock price, derivative price and cash account

Starting with a cash amount of Π_0^0 , the portfolio composed of both stock and cash is thus given by:

$$\begin{aligned}\Pi_0^0 &= \Delta_0^0 S_0^0 + (\Pi_0^0 - \Delta_0^0 S_0^0) \\ &= \Delta_0^0 S_0^0 + B_0^0\end{aligned}\tag{31}$$

Where Δ_0^0 gives the number of shares to hold at node $(0, 0)$ and B_0^0 the amount of cash held at node $(0, 0)$.

Given the self-financing nature of the portfolio, the only changes in value are due to the stock appreciation/depreciation and the interests gained at the bank:

$$\Pi_1 = \Delta_0 S_1 + e^r B_0\tag{32}$$

Our goal is to have a portfolio Π_i^j equal to the derivative price V_i^j whatever happens. With the simple 1-period CRR, this is translated by having both $\Pi_1^0 = V_1^0$ and $\Pi_1^1 = V_1^1$:

$$\begin{cases} \Delta_0^0 S_1^0 + e^r B_0^0 = V_1^0 \\ \Delta_0^0 S_1^1 + e^r B_0^0 = V_1^1 \end{cases}\tag{33}$$

Solving this simple system of two equations with two unknowns, one finds that:

$$\Delta_0^0 = \frac{V_1^0 - V_1^1}{S_0^0(u - d)} \quad B_0^0 = e^{-r} \left(\frac{uV_1^1 - dV_1^0}{u - d} \right)\tag{34, 35}$$

Plugging these into equation (31), one can find after some re-arranging the **no-arbitrage price of a derivative** paying $\psi(T)$ at time 1 (Vrins, 2020):

$$V_0^0 := \Pi_0^0 = e^{-r} \left(V_1^0 \frac{e^r - d}{u - d} + V_1^1 \frac{u - e^r}{u - d} \right)\tag{36}$$

[\[Return to 2.5 Replication strategy - Cox-Ross-Rubinstein model\]](#)

8.3 CRR pricing formula through risk-neutral

Let us now illustrate in the CRR model how the no-arbitrage price developed in equation (36), $V_0^0 := \Pi_0^0 = e^{-r} (V_1^0 \frac{e^r - d}{u - d} + V_1^1 \frac{u - e^r}{u - d})$, can be found again in risk-neutral pricing thanks to the FTAP.

Considering the continuously compounding discount rate $\beta_t = e^{rt}$, the discounted stock $\tilde{S}_t = \frac{S_t}{\beta_t}$ and the independent log-returns of the stock $Y_{t+1} = \frac{S_{t+1}}{S_t}$, we write the $\mathbb{Q}$ -expectation of the discounted stock price (Vrins, 2020):

$$\begin{aligned} \mathbb{E}^{\mathbb{Q}} [\tilde{S}_{t+1} | \mathcal{F}(t)] &= \frac{1}{\beta_{t+1}} \mathbb{E}^{\mathbb{Q}} [Y_{t+1} S_t | \mathcal{F}(t)] \\ &= \frac{S_t}{\beta_{t+1}} \mathbb{E}^{\mathbb{Q}} [Y_{t+1}] \\ &= \frac{S_t}{\beta_t e^r} \frac{\cancel{S_t}(\tilde{p}u + (1 - \tilde{p})d)}{\cancel{S_t}} \\ &= \tilde{S}_t \frac{\tilde{p}u + (1 - \tilde{p})d}{e^r} \end{aligned} \tag{37}$$

Which can be written given that the risk-free rate is known and that the log-returns are independent from one another.

At first view, the discounted stock price process has no reason to be a martingale. However, the FTAP states that there exists **an unique $\mathbb{Q}$ measure** for which that it is the case. Let us thus seek the precise $\tilde{p}$ value that will ensure the martingale property of the discounted stock price, i.e., $\frac{\tilde{p}u + (1 - \tilde{p})d}{e^r} = 1$.

It is easily found to be:

$$\tilde{p} = \frac{e^r - d}{u - d} \tag{38}$$

Note that $\tilde{p}$ is indeed a probability, i.e. it is bounded by 0 and 1, as long as $0 < d < e^r < u$, which was given in equation (9) as one of the assumptions of the CRR model! Thus, $\mathbb{Q}$ is indeed a probability measure and by assumption equivalent to $\mathbb{P}$.

Most importantly, we finally observe the FTAP in action: the no-arbitrage price found in equation (36) corresponds perfectly to the conditional expectation of the process under the equivalent martingale measure, defined as the measure turning the discounted stock price into a martingale (Vrins, 2020)!

Equation (36) can be equivalently written as:

$$\begin{aligned} V_0^0 &:= \Pi_0^0 = \mathbb{E}^{\mathbb{Q}} [\psi(S_1)e^{-r}] = e^{-r} \left(V_1^0 \frac{e^r - d}{u - d} + V_1^1 \frac{u - e^r}{u - d} \right) \\ &= e^{-r} (V_1^0 \tilde{p} + V_1^1 \tilde{q}) \end{aligned} \quad (39)$$

Finally, in a CRR model, we have showed how and why the cost of the self-financing replicating portfolio is perfectly equal to the derivative price by replication or risk-neutral thanks to the Fundamental Theorem of Asset Pricing.

Note that we have shown that the discounted stock price is a martingale under the risk-neutral measure only for consecutive steps t and $t + 1$. This can be generalized for any steps $s < t$ using the tower law with $t \geq s + 1$ (Vrins, 2020).

Finally, up until now we have been working in discrete time with the CRR model. The results we have shown so far are naturally also valid in continuous time. Indeed, while it is out of scope of this work, it is interesting to know that the link between the Black-Scholes-Merton and the risk-neutral expectation is given by the Feynman-Kac theorem.

The following explanations of the Feynman-Kac theorem are completely taken from Prof. Vrins course at LSM (Vrins, 2020).

How can we explain that the function $f(t, x)$ solving the BSM PDE along the terminal conditional $f(T, x) = \psi(x)$ takes the well-known form of a conditional expectation under the risk-neutral measure $e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [\psi(S_T) | S_t = x]$?

It is based on the Feynman-Kac theorem. Considering the following PDE with terminal conditional $f(T, x) = \psi(x)$:

$$f_t(t, x) + \alpha(t, x)f_x(t, x) + \frac{\gamma^2(t, x)}{2}f_{xx}(t, x) - v(t, x)f(t, x) + g(t, x) = 0 \quad (40)$$

Then, defining $\beta_t := e^{\int_0^t v(X_s, s) ds}$, the solution of this PDE is given by:

$$f(t, x) = \mathbb{E}^{\mathbb{Q}} \left[\int_t^T \frac{\beta_u}{\beta_T} g(u, X_u) du + \frac{\beta_t}{\beta_T} \psi(X_T) | X_t = x \right], dX_t = \alpha(t, X_t)dt + \gamma(t, X_t)dW_t^{\mathbb{Q}} \quad (41)$$

Note that the BSM PDE is a special case where T is the maturity, ψ is the payoff and

$$\alpha(t, x) = rx, \gamma(t, x) = \sigma x, v(t, x) = r, g(t, x) = 0 \quad (42)$$

Applying Feynman-Kac to it, the time- t price of the option is $V_t = f(t, X_t)$ where:

$$f(t, x) = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [\psi(X_T) | X_t = x] \quad (43)$$

and X solves $dX_t = rX_t dt + \sigma X_t dW_t^{\mathbb{Q}}$.

In other words, thanks to the Feynman-Kac theorem, V_t is the conditional risk-neutral expectation of the discounted payoff where the underlying asset X follows a geometric Brownian motion.

[\[Return to 2.4 Risk-neutral pricing formulas - Cox-Ross-Rubinstein model\]](#)

8.4 CRR option pricing examples

Under the pricing measure, the price at all nodes is simply the discounted value of the two children nodes. The price tree is therefore filled backwards, starting from the leaves (i.e. the payoff) (Vrins, 2020).

European options

Reminding the payoff of European call options with $i = T$,

$$\psi_i = \max(0, S_i - K) \quad (44)$$

The stock price tree is simulated with the following inputs: $S_0^0 = 100, u = 2, d = \frac{1}{2}, r = 10\%, K = 100$. Thus, $\tilde{p} = 40\%$. δ is assumed to be equal to 1.

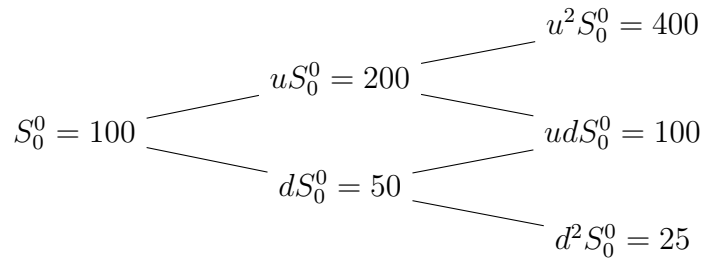


Figure 15: European option CRR: 3-period stock tree

And then start pricing the option by filling the tree backwards, starting from the leaves.

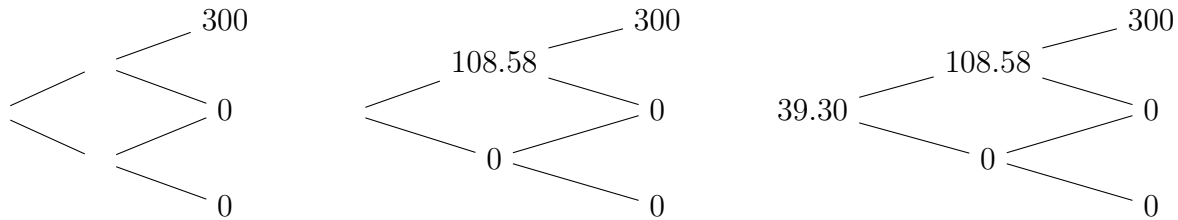


Figure 16: European option CRR: 3-period option pricing tree

The non-null computations are:

- node(2, 0) : $\max(0, 400 - 100) = 300$
- node(1, 0) : $e^{-0.1}(40\%300) \approx 108.58$
- node(0, 0) : $e^{-0.1}(40\%108.58) \approx 39.30$

Asian options

To be adapted to the CRR model, let us define $Y_i = \sum_{k=0}^i S_k$ and $V_i(S_i, Y_i)$ be the option price. We assume that the fixing dates are the tree nodes. The Asian call option payoff at node $i = T$ with the starting node at $i = 0$ will be:

$$\psi_i(S_i, Y_i) = \max(0, \frac{1}{i+1} Y_i - K) \quad (45)$$

Assuming the same variables values as before, i.e. $S_0 = 100, u = 2, d = \frac{1}{2}, r = 10\%, K = 100, \tilde{p} = 40\%, \delta = 1$. Starting with the stock and Y simulation,

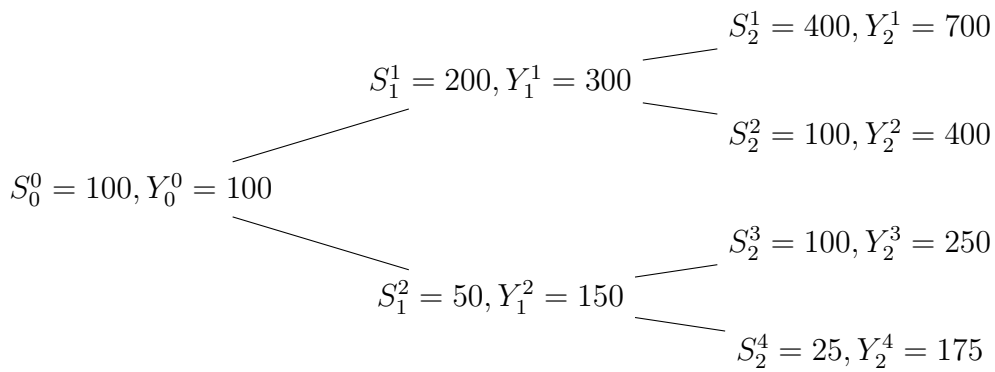


Figure 17: Asian option CRR: 3-period stock/sum tree

The call option is priced by filling the tree backwards, in the same manner as earlier:

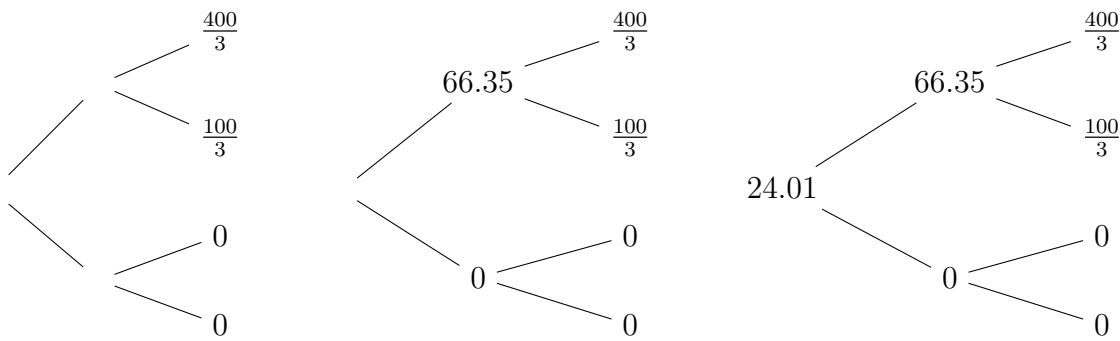


Figure 18: Asian option CRR: 3-period option pricing tree

The non-null computations are:

- $\text{node}(2, 1) : (\frac{1}{3}700 - 100)^+ = \frac{400}{3}$
- $\text{node}(2, 2) : (\frac{1}{3}400 - 100)^+ = \frac{100}{3}$
- $\text{node}(1, 1) : e^{-0.1}(40\%\frac{400}{3} + 60\%\frac{100}{3}) \approx 66.35$
- $\text{node}(0, 0) : e^{-0.1}(40\%66.35) \approx 24.01$

[\[Return to 2.4 Risk-neutral pricing formulas - Cox-Ross-Rubinstein model\]](#)

8.5 CRR option hedging examples

European options

Re-using the same stock simulation and call option price as in **Figures 15** and **16**, with $S_0 = 100$, $u = 2$, $d = \frac{1}{2}$, $r = 10\%$ and $K = 100$. Thus, $\tilde{p} = 40\%$. Note that for simplicity it is assumed that $\delta = 1$.

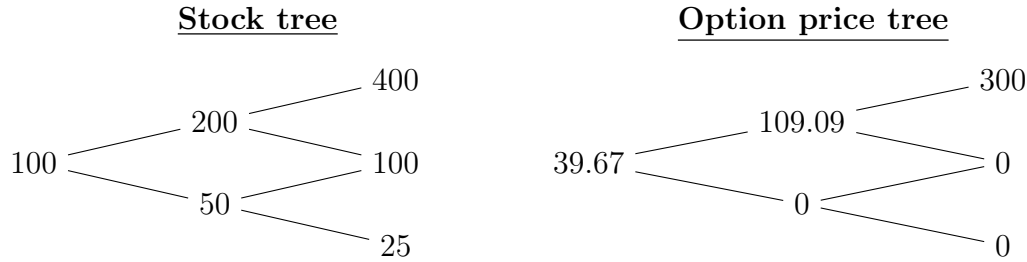


Figure 19: European option CRR: (left) Stock simulation & (right) Option price trees (taken from figures 15 and 16)

Using the formula $\Delta_i^j = \frac{V_{i+1}^j - V_{i+1}^{j+1}}{S_i^j(u-d)}$ derived earlier, let us start creating the self-financing replicating portfolio:

We start with $\Pi_0^0 = V_0^0$, and buy Δ_0^0 shares of stock, leaving the cash account with $\Pi_0^0 - \Delta_0^0 S_0^0$. The tree is computed in a forward manner, i.e., the opposite of risk-neutral pricing.

Finally, in the trees below, each node contains two elements: the value **before** and **after** re-balancing the accounts. The first tree has under-braces to illustrate it.

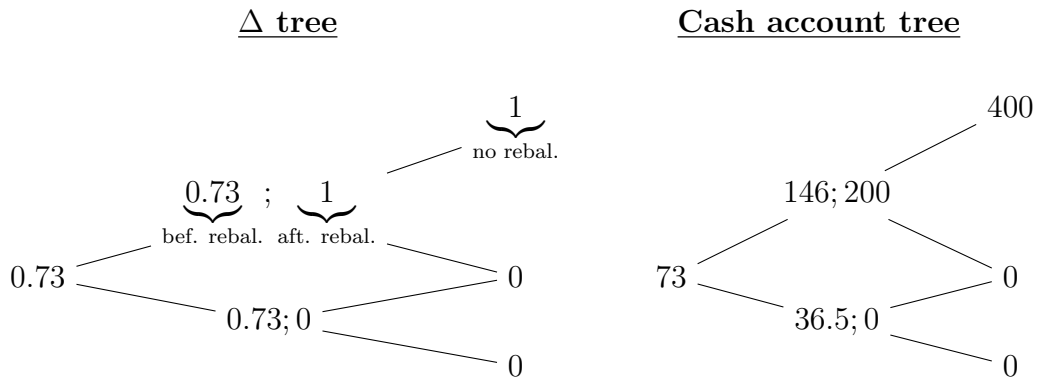


Figure 20: European option CRR: (left) Number of shares to hold Δ & (right) Equity account trees

The non-null computations are:

- $\text{node}(0, 0) : \frac{109.09-0}{200-50} \approx 0.73$
- $\text{node}(1, 0):$
 - before balancing: 0.73
 - after re-balancing: $\frac{300-0}{400-100} = 1$
- $\text{node}(1, 1):$
 - before balancing: 0.73
 - after re-balancing: 0
- $\text{node}(2, 0) : 1$ (no re-balancing!)
- $\text{node}(0, 0) : 0.73 \times 100 = 73$
- $\text{node}(1, 0):$
 - before balancing: $0.73 \times 200 = 146$
 - after re-balancing: 1×200
- $\text{node}(1, 1):$
 - before balancing: $0.73 \times 50 = 36.5$
 - after re-balancing: 0
- $\text{node}(2, 0) : 1 \times 400$

Cash account tree:

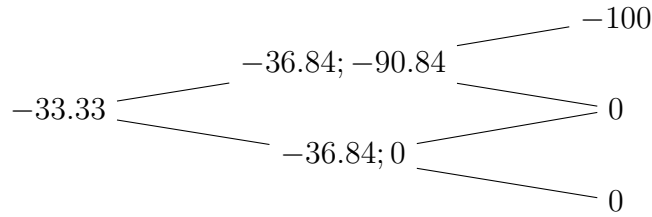


Figure 21: European option CRR: Cash account tree

The non-null computations are:

- $\text{node}(0, 0) : 39.67 - 73 = -33.33$
- $\text{node}(1, 0):$
 - before balancing: $-33.33e^{0.1} \approx -36.84$
 - after re-balancing: $\underbrace{(-36.84 + 146)}_{\Pi \text{ bef. rebal.}} - 200 = -90.84$
- $\text{node}(1, 1):$
 - before balancing: $-33.33e^{0.1} \approx -36.84$
 - after re-balancing: 0
- $\text{node}(2, 0) : -90.84e^{0.1} \approx -100$ (no re-balancing! Simply the interests)

Portfolio tree:

Summing the cash and equity accounts, observe that the portfolio and option have the same values.

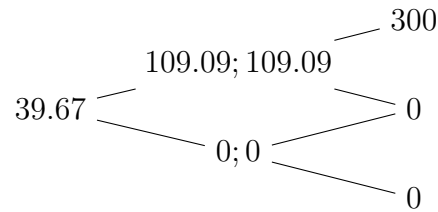


Figure 22: European option CRR: Replicating Portfolio tree

The non-null computations are:

- node(0, 0) : $-33.33 + 73 = 39.67$
- node(1, 0):
 - before balancing: $-36.84 + 146 \approx 109.09$
 - after re-balancing: $-90.84 + 200 \approx 109.09$
- node(1, 1):
 - before balancing: $-36.84 + 36.5 \approx 0$
 - after re-balancing: $0 + 0 = 0$
- node(2, 0) : $-100 + 400 = 300$

By creating a self-financing replicating portfolio, we have proved that that option price is unique and therefore arbitrage-free.

Asian options

Re-using the same stock simulation and call option price as in **Figures 17** and **18**, with $S_0 = 100, u = 2, d = \frac{1}{2}, r = 10\%$ and $K = 100$. Thus, $\tilde{p} = 40\%$. We assume that the fixing dates are the tree nodes. Note that for simplicity it is assumed that $\delta = 1$.

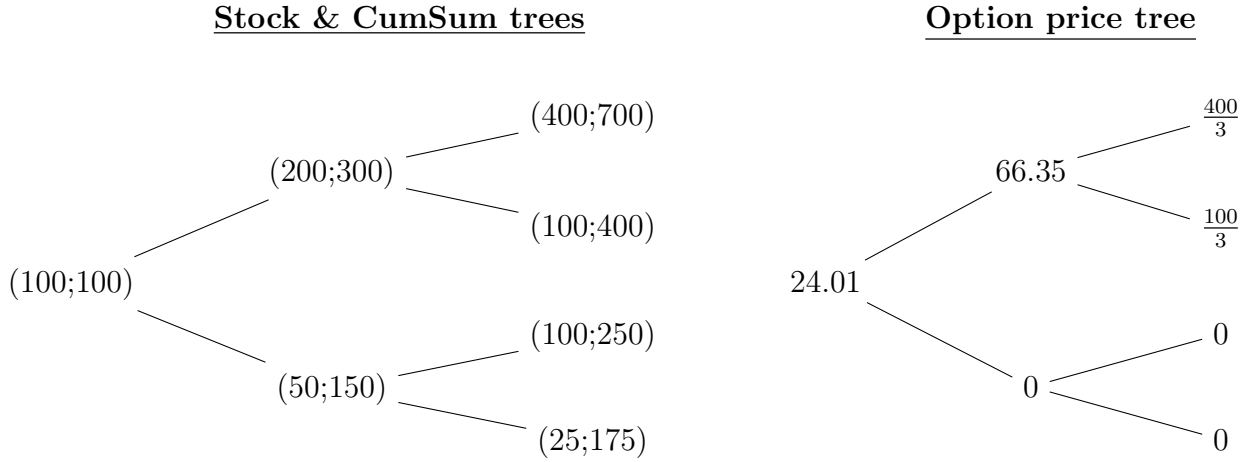


Figure 23: Asian option CRR: (left) (Stock; Cum Sum) simulation & (right) Option price trees (taken from figures 17 and 18)

For Asian options, the Δ formula is the same as European options. We simply change its notation to adapt it for this derivative:

$$\Delta_i(S_i, Y_i) = \frac{V_{i+1}(uS_i, Y_i + uS_i) - V_{i+1}(dS_i, Y_i + dS_i)}{S_i(u - d)} \quad (46)$$

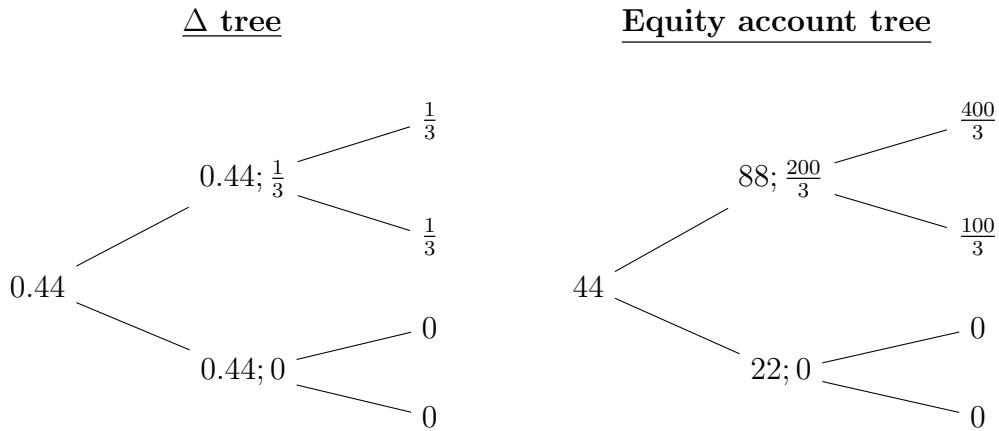


Figure 24: Asian option CRR: (left) Number of shares to hold Δ & (right) Equity account trees

The non-null computations are:

- $\text{node}(0, 0) : \frac{66.35-0}{200-50} = 0.44$
- $\text{node}(1, 1):$
 - before balancing: 0.44
 - after re-balancing: $\frac{\frac{400}{3}-\frac{100}{3}}{400-100} = \frac{1}{3}$
- $\text{node}(1, 2):$
 - before balancing: 0.44
 - after re-balancing: 0
- $\text{node}(2, 1) : \frac{1}{3}$ (no re-balancing!)
- $\text{node}(2, 2) : \frac{1}{3}$ (no re-balancing!)
- $\text{node}(0, 0) : 0.44 * 100 = 44$
- $\text{node}(1, 1):$
 - before balancing: $0.44 * 200 = 88$
 - after re-balancing: $\frac{200}{3}$
- $\text{node}(1, 2):$
 - before balancing: $0.44 * 50 = 22$
 - after re-balancing: 0
- $\text{node}(2, 1) : \frac{400}{3}$
- $\text{node}(2, 2) : \frac{100}{3}$

Cash account tree

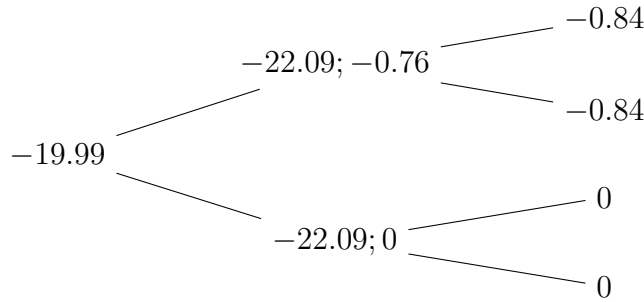


Figure 25: Asian option CRR: Cash account tree

- $\text{node}(0, 0) : 24.01 - 44 = -19.99$
- $\text{node}(1, 1):$
 - before balancing: $-19.99e^{0.1} = -22.09$
 - after re-balancing: $(-22.09 + 88) - \frac{200}{3} = -0.76$
- $\text{node}(1, 2):$
 - before balancing: $-19.99e^{0.1} = -22.09$
 - after re-balancing: 0
- $\text{node}(2, 1) : -0.76e^{0.1} = -0.84$ (no re-balancing! Simply the interests)
- $\text{node}(2, 2) : -0.76e^{0.1} = -0.84$ (no re-balancing! Simply the interests)

Portfolio tree

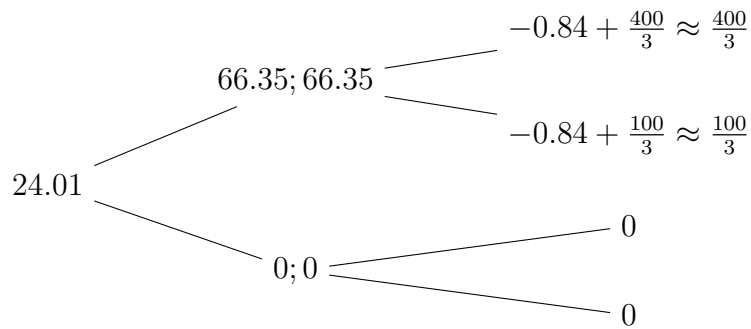


Figure 26: Asian option CRR: Replicating Portfolio tree

- node(0, 0) : $-19.99 + 44 = 24.01$
- node(1, 1):
 - before balancing: $-22.09 + 88 \approx 66.35$
 - after re-balancing: $-0.76 + \frac{200}{3} \approx 66.35$
- node(1, 2):
 - before balancing: $-22.09 + 22 \approx 0$
 - after re-balancing: $0 + 0 = 0$
- node(2, 1) : $-0.84 + \frac{400}{3} \approx \frac{400}{3}$
- node(2, 2) : $-0.84 + \frac{100}{3} \approx \frac{100}{3}$

[\[Return to 2.5 Replication strategy - Cox-Ross-Rubinstein model\]](#)

8.6 BSM European option risk-neutral price full derivation

To get the price of an European-type derivative, one can either solve the BSM PDE given in equation (12) with terminal condition the payoff of the desired European derivative, or use risk-neutral pricing (Shreve, 2004b). The former requiring intensive stochastic calculus which is out of the scope of the thesis, only the latter is covered.

As explained in earlier sections, under the risk-neutral setting, the price of any asset V_t is its discounted payoff under the risk-neutral measure $\mathbb{Q}$.

Starting with European calls, their price is given by:

$$\begin{aligned} V_t &= \beta_t \mathbb{E}^{\mathbb{Q}} \left[\frac{(S_T - K)^+}{\beta_T} | \mathcal{F}(t) \right] \\ &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [(S_T - K)^+ | \mathcal{F}(t)] \end{aligned} \quad (47)$$

Where $\beta_t = e^{rt}$, and r is the constant risk-free rate.

To compute the expectation, the dynamics of the stock price S under $\mathbb{Q}$ must be known. Its SDE is the geometric Brownian motion $dS_t = \mu S_t dt + \sigma S_t dW_t$ (from equation (13)) and must be solved.

Stock dynamics under $\mathbb{P}$

We first derive the dynamics under $\mathbb{P}$ with Ito's lemma, which is given by:

$$df(t, X_t) = f_t(t, X_t)dt + f_x(t, X_t)dX_t + \frac{1}{2}f_{xx}(t, X_t)(dX_t)^2 \quad (48)$$

For conciseness, let us write $f_t = f_t(t, X_t)$, $f_x = f_x(t, X_t)$, $f_{xx} = f_{xx}(t, X_t)$.

Using dummy function $f(t, x) = \ln(x)$, its partial derivatives are computed:

$$\begin{aligned} f_t &= \frac{\partial f(t, x)}{\partial t} = 0 \\ f_x &= \frac{\partial f(t, x)}{\partial x} = \frac{1}{x} \\ f_{xx} &= \frac{\partial f(t, x)}{\partial x^2} = -\frac{1}{x^2} \end{aligned} \quad (49)$$

It is found that,

$$d\ln(S_t) = \frac{1}{S_t}dS_t - \frac{1}{2S_t^2}dS_t^2 \quad (50)$$

Substituting dS_t with its SDE expression,

$$\begin{aligned} d\ln(S_t) &= \frac{1}{S_t}(\mu S_t dt + \sigma S_t dW_t) - \frac{1}{2S_t^2}(\mu S_t dt + \sigma S_t dW_t)^2 \\ &= \mu dt + \sigma dW_t - \frac{1}{2S_t^2}[\underbrace{\mu^2 S_t^2 dt^2}_{=0} + 2\mu\sigma S_t^2 \underbrace{dt dW_t}_{=0} + \sigma^2 S_t^2 \underbrace{dW_t^2}_{=dt}] \\ d\ln(S_t) &= \mu dt + \sigma dW_t - \frac{\sigma^2}{2}dt \end{aligned} \quad (51)$$

Given that $(dt)^2 = dt dW_t = 0$ and $(dW_t)^2 = dt$.

Then, integrate both sides of the equation from 0 to t ,

$$\begin{aligned} \ln(S_t) - \ln(S_0) &= \mu t + \sigma(W_t - \underbrace{W_0}_{=0}) - \frac{\sigma^2}{2}t \\ \ln\left(\frac{S_t}{S_0}\right) &= \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t \\ \frac{S_t}{S_0} &= e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \\ S_t &= S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \end{aligned} \quad (52)$$

Where $\int_0^t dW_t = W_t - W_0$ is the definition of an Itô integral, and that $W_0 = 0$ by properties of Brownian motions.

Stock dynamics under $\mathbb{Q}$

Then, when changing from the historical measure $\mathbb{P}$ to the risk-neutral measure $\mathbb{Q}$, the drift μ can be swapped for the risk-free rate r at no cost.

Let us show how. Note that this proof is mostly taken from Prof. Vrms course at LSM ([Vrms, 2020](#)). We first change the probability measure using a Radon-Nikodym derivative. Let W be a Brownian motion under $\mathbb{P}$ and define $\mathbb{Q}$ by using as Radon-Nikodym derivative the strictly positive and unit $\mathbb{P}$ -expectation random variable

$$L := \frac{d\mathbb{Q}}{d\mathbb{P}} = e^{\frac{\theta^2}{2}T - \theta W_T} \quad (53)$$

As such, we can define $\tilde{W}_t = W_t + \theta t$ a Brownian motion under $\mathbb{Q}$. From the solution to the GBM SDE, equation (52), we have

$$S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \quad (54)$$

Substituting W_t by $\tilde{W}_t - \theta t$,

$$\begin{aligned} S_t &= S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma(\tilde{W}_t - \theta t)} \\ &= S_0 e^{(\mu - \sigma\theta - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t} \end{aligned} \quad (55)$$

Discounting the solution, i.e. multiplying by e^{-rt}

$$\begin{aligned} &= S_0 e^{(\mu - \sigma\theta - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t} e^{-rt} \\ &= S_0 e^{(\mu - \sigma\theta - r - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t} \end{aligned} \quad (56)$$

Separating the exponentials,

$$\begin{aligned} &= S_0 e^{(\mu - \sigma\theta - r - \frac{\sigma^2}{2})t} e^{\sigma\tilde{W}_t} \\ &= S_0 e^{(\mu - \sigma\theta - r)t} \underbrace{e^{-\frac{\sigma^2}{2}t + \sigma\tilde{W}_t}}_{\text{exp. martingale}} \end{aligned} \quad (57)$$

Given the exponential martingale properties, we look for the specific θ that makes S_t a martingale, i.e. that cancels out $e^{(\mu - \sigma\theta - r)t}$

$$\begin{aligned} \mu - \sigma\theta - r &= 0 \\ \theta &= \frac{\mu - r}{\sigma} \end{aligned} \quad (58)$$

θ is called the market price of risk. Substituting it in the earlier equation (55),

$$\begin{aligned} S_t &= S_0 e^{(\mu - \sigma \frac{\mu - r}{\sigma} - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t} \\ &= S_0 e^{(\mu - \cancel{\sigma} \frac{\mu - r}{\cancel{\sigma}} - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t} \\ &= S_0 (\mu - \mu + r - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t \\ S_t &= S_0 (r - \frac{\sigma^2}{2})t + \sigma\tilde{W}_t \end{aligned} \quad (59)$$

This way, we have proved that replacing μ by the risk-free rate will make the discounted stock price under $\mathbb{Q}$ a martingale, and thus derived its dynamics its $\mathbb{Q}$ -dynamics.

Back to the risk-neutral expectation & solving the integral

Going back to equation (47), the European call price under the risk-neutral expectation becomes:

$$\begin{aligned}
 V_t &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} [(S_T - K)^+ | \mathcal{F}(t)] \\
 V_t &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} \left[(S_t e^{(r-\frac{\sigma^2}{2})(T-t) + \sigma(\tilde{W}_T - \tilde{W}_t)} - K)^+ | \mathcal{F}(t) \right] \\
 &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} \left[(S_t e^{(r-\frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z} - K)^+ | \mathcal{F}(t) \right] \\
 &= e^{-r(T-t)} \int_{-\infty}^{\infty} (S_t e^{(r-\frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z} - K)^+ f(Z) dZ
 \end{aligned} \tag{60}$$

Where $Z = -\frac{\tilde{W}_T - \tilde{W}_t}{\sqrt{T-t}}$, $Z \stackrel{\mathbb{Q}}{\sim} \mathcal{N}(0, 1)$ and its probability density function $f(Z) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}Z^2)$. Solving the integral gives the European call pricing formula found through risk-neutral expectation. Let us do so.

From there, let us get rid of the maximum:

$$\begin{aligned}
 S_t e^{(r-\frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z} - K &> 0 \\
 e^{(r-\frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z} &> K/S_t \\
 (r - \frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z &> \ln(K/S_t) \\
 Z &< \frac{\ln(S_t/K) + (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}
 \end{aligned} \tag{61}$$

The right-hand element of the inequality will therefore be our upper bound of the integral in equation (??). Let us call it d_2 .

$$\begin{aligned}
 V_t &= e^{-r(T-t)} \int_{-\infty}^{d_2} (S_t e^{(r-\frac{\sigma^2}{2})(T-t) - \sigma\sqrt{T-t}Z} - K) f(Z) dZ \\
 &= e^{-r(T-t)} \left(\underbrace{S_t e^{(r-\frac{\sigma^2}{2})(T-t)} \int_{-\infty}^{d_2} e^{-\sigma\sqrt{T-t}Z} f(Z) dZ}_{\text{blue line}} - \underbrace{K \int_{-\infty}^{d_2} f(Z) dZ}_{\text{green line}} \right)
 \end{aligned} \tag{62}$$

Starting with the blue integral, by the definition of the PDF of a Normal $\mathcal{N}(0, 1)$,

$$\begin{aligned} \underline{S_t e^{(r - \frac{\sigma^2}{2})(T-t)} \int_{-\infty}^{d_2} e^{-\sigma\sqrt{T-t}Z} f(Z) dZ} &= \frac{S_t e^{(r - \frac{\sigma^2}{2})(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2} e^{-\sigma\sqrt{T-t}Z} e^{-\frac{Z^2}{2}} dZ \\ &= \frac{S_t e^{(r - \frac{\sigma^2}{2})(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2} e^{-\sigma\sqrt{T-t}Z + \frac{\sigma^2}{2}(T-t) - \frac{\sigma^2}{2}(T-t)} e^{-\frac{Z^2}{2}} dZ \end{aligned} \quad (63)$$

Taking out $e^{\frac{\sigma^2}{2}(T-t)}$ from the integral, then factorizing $-\frac{1}{2}$ inside the exponential, the following remarkable product can be formed: $-\frac{(Z + \sigma\sqrt{T-t})^2}{2} = -\frac{1}{2} [Z^2 + 2\sigma\sqrt{T-t}Z + \sigma^2(T-t)] = -\frac{Z^2}{2} - \sigma\sqrt{T-t}Z - \frac{\sigma^2}{2}(T-t)$

$$\begin{aligned} &= \frac{e^{\frac{\sigma^2}{2}(T-t)} S_t e^{(r - \frac{\sigma^2}{2})(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2} e^{-\sigma\sqrt{T-t}Z - \frac{\sigma^2}{2}(T-t) - \frac{Z^2}{2}} dZ \\ &= \frac{S_t e^{r(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2} e^{\frac{-1}{2}(2\sigma\sqrt{T-t}Z + \sigma^2(T-t) + Z^2)} dZ \\ &= \frac{S_t e^{r(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2} e^{-\frac{(Z + \sigma\sqrt{T-t})^2}{2}} dZ \end{aligned} \quad (64)$$

Substituting $y = Z + \sigma\sqrt{T-t}$, the new integral borders are $-\infty$ and $d_2 + \sigma\sqrt{T-t}$.

$$\begin{aligned} &= \frac{S_t e^{r(T-t)}}{\sqrt{2\pi}} \int_{-\infty}^{d_2 + \sigma\sqrt{T-t}} e^{-\frac{Y^2}{2}} dY \\ &= S_t e^{r(T-t)} \mathbb{P}[Z \leq d_2 + \sigma\sqrt{(T-t)}] \\ &= S_t e^{r(T-t)} \Phi(d_2 + \sigma\sqrt{(T-t)}) \end{aligned} \quad (65)$$

Define $d_1 = d_2 + \sigma\sqrt{T}$ and solve the green integral in a similar manner, obtaining $K\Phi(d_2)$. Hence,

$$\begin{aligned} V_t &= e^{-r(T-t)} [S_t e^{r(T-t)} \Phi(d_1) - K\Phi(d_2)] \\ &= S_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2) \end{aligned} \quad (66)$$

Which is the Black-Scholes-Merton European call pricing formula found through risk-neutral valuation.

For put options, the earlier proof can be used but it is much quicker to compute the pricing formula using the Put-Call parity $P_t + S_t = C_t + Ke^{-r(T-t)}$:

$$\begin{aligned}
 P_t + S_t &= C_t + Ke^{-r(T-t)} \\
 P_t &= S_t\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2) + Ke^{-r(T-t)} - S_t \\
 &= Ke^{-r(T-t)}(1 - \Phi(d_2)) - S_t(1 - \Phi(d_1)) \\
 &= Ke^{-r(T-t)}\Phi(-d_2) - S_t\Phi(-d_1)
 \end{aligned} \tag{67}$$

[\[Return to 2.4 Risk-neutral pricing formulas - European options\]](#)

8.7 BSM European option Δ full derivation

In this appendix, we solve for $\Delta := \frac{dV_t}{dS_t}$ and also for the other Greeks.

For these derivations, some preliminary equations are needed:

$$\begin{aligned} d_2(t, x) &= d_1(t, x) - \sigma\sqrt{T-t} \\ \frac{\partial d_2(t, x)}{\partial x} &= \frac{\partial d_1(t, x)}{\partial x} \\ \frac{\partial d_2(t, x)}{\partial t} &= \frac{\partial d_1(t, x)}{\partial t} + \frac{\sigma}{2\sqrt{T-t}} \end{aligned} \tag{68}$$

And from (Yu & Xie, 2013),

$$\begin{aligned} \frac{\phi(d_1)}{\phi(d_2)} &= \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d_1^2}}{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}d_2^2}} \\ &= e^{-\frac{1}{2}d_1^2 + \frac{1}{2}d_2^2} \\ &= e^{\frac{1}{2}[d_2^2 - d_1^2]} \end{aligned} \tag{69}$$

Developing the remarkable products, it can be found that

$$\begin{aligned} \frac{\phi(d_1)}{\phi(d_2)} &= e^{\frac{1}{2}(-\sigma\sqrt{T-t})(2d_1 - \sigma\sqrt{T-t})} \\ &= e^{\frac{-\sigma\sqrt{T-t}}{2} \left[2 \frac{\ln(S/K) + (T-t)(r + \frac{\sigma^2}{2})}{\sigma\sqrt{T-t}} - \sigma\sqrt{T-t} \right]} \\ &= e^{-\ln(S/K) - (T-t)(r + \frac{\sigma^2}{2}) + \sigma^2(T-t)} \\ &= e^{\ln(K/S) - r(T-t)} \\ &= \frac{K}{S} e^{-r(T-t)} \end{aligned} \tag{70}$$

Applying cross product rule,

$$S\phi(d_1) = K e^{-r(T-t)} \phi(d_2) \tag{71}$$

Let us find the option Greeks:

$$1. \Delta = f_x(t, S_t) = \left. \frac{\partial f(t, x)}{\partial x} \right|_{(t, x) = (t, S_t)}$$

$$\begin{aligned} \Delta &= \frac{\partial [x\Phi(d_1(t, x)) - Ke^{-r(T-t)}\Phi(d_2(t, x))]}{\partial x} \\ &= \Phi(d_1(t, x)) + x\phi(d_1(t, x))\frac{\partial d_1(t, x)}{\partial x} - Ke^{-r(T-t)}\phi(d_2(t, x))\frac{\partial d_2(t, x)}{\partial x} \end{aligned} \quad (72)$$

Additionally, from the preliminary equations, we know that $\frac{\partial d_1(t, x)}{\partial x} = \frac{\partial d_2(t, x)}{\partial x}$ and $x\phi(d_1(t, x)) = Ke^{-r(T-t)}\phi(d_2(t, x))$.

$$\Delta = \Phi(d_1(t, x)) \quad (73)$$

$$2. \Gamma = f_{xx}(t, S_t) = \frac{\partial \Delta}{\partial x}$$

$$\Gamma = \phi(d_1(t, x))\frac{\partial d_1(t, x)}{\partial x} \quad (74)$$

$$\Gamma = \frac{1}{\sigma S \sqrt{T-t}} \phi(d_1(t, x)) \quad (75)$$

$$3. \theta = f_t(t, S_t) = \left. \frac{\partial f(t, x)}{\partial t} \right|_{(t, x) = (t, S_t)}$$

$$\begin{aligned} \theta &= \frac{\partial [x\Phi(d_1(t, x)) - Ke^{-r(T-t)}\Phi(d_2(t, x))]}{\partial t} \\ &= x\phi(d_1(t, x))\frac{\partial d_1(t, x)}{\partial t} - \left[rKe^{-r(T-t)}\Phi(d_2(t, x)) + Ke^{-r(T-t)}\phi(d_2(t, x))\frac{\partial d_2(t, x)}{\partial t} \right] \end{aligned} \quad (76)$$

Again from preliminary equations, we know that $\frac{\partial d_2(t,x)}{\partial t} = \frac{\partial d_1(t,x)}{\partial t} + \frac{\sigma}{2\sqrt{T-t}}$.

$$\begin{aligned}
 \theta &= x\phi(d_1(t,x)) \frac{\partial d_1(t,x)}{\partial t} - \\
 &\quad \left[rKe^{-r(T-t)}\Phi(d_2(t,x)) + Ke^{-r(T-t)}\phi(d_2(t,x)) \left[\frac{\partial d_1(t,x)}{\partial t} + \frac{\sigma}{2\sqrt{T-t}} \right] \right] \\
 &= \frac{\partial d_1(t,x)}{\partial t} [x\phi(d_1(t,x)) - Ke^{-r(T-t)}\phi(d_2(t,x))] - rKe^{-r(T-t)}\Phi(d_2(t,x)) \\
 &\quad - Ke^{-r(T-t)}\phi(d_2(t,x)) \frac{\sigma}{2\sqrt{T-t}}
 \end{aligned} \tag{77}$$

From the preliminary equations, $x\phi(d_1(t,x)) = Ke^{-r(T-t)}\phi(d_2(t,x))$.

$$\theta = -rKe^{-r(T-t)}\Phi(d_2(t,x)) - \frac{\sigma}{2\sqrt{T-t}}x\phi(d_1(t,x)) \tag{78}$$

[\[Return to 2.5 Replication strategy - European options\]](#)

8.8 BSM European option replicating strategy

Let us start by inferring the dynamics of the option price by applying Ito's lemma to the BSM PDE V_t given in equation (12). Applying one variable Ito's lemma to $V_t = f(t, S_t)$:

$$df(t, S_t) = f_t(t, S_t)dt + f_x(t, S_t)dS_t + \frac{1}{2}f_{xx}(t, S_t)(dS_t)^2 \quad (79)$$

Simplifying the notation and applying to our case,

$$dV_t = \left(f_t + \frac{1}{2}\sigma^2 S_t^2 f_{xx} \right) dt + f_x dS_t \quad (80)$$

Let us continue by creating a portfolio Π_t that will replicate at all $t \in [0; T]$ the derivative price. At time t , invest in Δ_t shares. The portfolio will therefore be composed of the value of Δ_t shares and the cash reminder (Vrins, 2020).

$$\Pi_t = \underbrace{\Pi_t - \Delta_t S_t}_{\text{cash account}} + \underbrace{\Delta_t S_t}_{\text{equity account}} \quad (81)$$

Given the self-financing assumption of the portfolio, the cash account grows at the risk-free rate and the equity account changes are only due to the stock appreciation/depreciation:

$$d\Pi_t = r(\Pi_t - \Delta_t S_t)dt + \Delta_t dS_t \quad (82)$$

Let us now equate (82) to (80):

$$d\Pi_t = dV_t \quad (83)$$

$$r(\Pi_t - \Delta_t S_t)dt + \Delta_t dS_t = \left(f_t(t, S_t) + \frac{1}{2}\sigma^2 S_t^2 f_{xx}(t, S_t) \right) dt + f_x(t, S_t)dS_t \quad (84)$$

Observe that if:

$$\begin{cases} r(\Pi_t - \Delta_t S_t) = f_t + \frac{1}{2}\sigma^2 S_t^2 f_{xx} \\ \Delta_t = f_x \end{cases} \quad (85)$$

Then $\Pi_t = V_t$. In other words, as long as the strategy holds $\Delta_t = f_x = \nu\Phi(\nu d_1)$ ($\nu = 1$ for calls and -1 for puts) at all times t and re-balance the portfolio accordingly, it can perfectly replicate the BSM pricing formula. This is called the delta-hedge portfolio, as it hedges away the randomness embedded in S_t . Note that the Greeks derivations, i.e. f_t, f_{xx} and most importantly f_x are given in the previous appendix.

[\[Return to 2.5 Replication strategy - European options\]](#)

8.9 BSM Exchange option risk-neutral price full derivation

Starting with two correlated stocks following geometric Brownian motions,

$$\begin{aligned} dS_t^{(1)} &= \mu^{(1)} S_t^{(1)} dt + \sigma^{(1)} S_t^{(1)} dW_t^{(1)} \\ dS_t^{(2)} &= \mu^{(2)} S_t^{(2)} dt + \sigma^{(2)} S_t^{(2)} dW_t^{(2)} \end{aligned} \quad (86)$$

Where $dW_t^{(1)} dW_t^{(2)} = \rho dt$ where ρ is the Pearson's correlation coefficient of the Brownian motions of the two stock, allowing us to define $\sigma = \sqrt{(\sigma^{(1)})^2 + (\sigma^{(2)})^2 - 2\sigma^{(1)}\sigma^{(2)}\rho}$. Assume no dividends are paid.

Then, back to the discounted risk-neutral expectation of the payoff,

$$\begin{aligned} V_t &= \beta_t \mathbb{E}^{\mathbb{Q}} \left[\frac{(S_T^{(1)} - S_T^{(2)})^+}{\beta_T} \middle| \mathcal{F}_t \right] \\ &= \frac{\beta_t}{\beta_T} \mathbb{E}^{\mathbb{Q}} \left[(S_T^{(1)} - S_T^{(2)})^+ \middle| \mathcal{F}_t \right] \end{aligned} \quad (87)$$

The change of numéraire technique will now be used. Its purpose is to price financial products relatively to another tradable asset whilst preserving the Martingale property of the discounted payoff under $\mathbb{Q}$ (Shreve, 2004b)(Vrins, 2020).

Starting with defining the numéraires $M_t = \beta_t$ and $N_t = S_t^{(2)}$, the theorem states that there exists a measure $\mathbb{Q}^N$ equivalent to $\mathbb{Q}$ such that prices of assets expressed with respect to numéraire N_t are martingales under $\mathbb{Q}^N$ (Azizieh, 2017). It is obtained as such:

$$\frac{d\mathbb{Q}^N}{d\mathbb{Q}} = \frac{M_0 N_T}{N_0 M_T} \quad (88)$$

and for all random variable $Y - \mathcal{F}(T)$ measurable,

$$\mathbb{E}^{\mathbb{Q}^N} [Y | \mathcal{F}(t)] = \frac{M_t}{N_t} \mathbb{E}^{\mathbb{Q}} [Y \frac{N_T}{M_T} | \mathcal{F}(t)] \quad (89)$$

Going back to equation (87), we do some re-arranging and then use $S_t^{(2)}$ as numéraire:

$$\begin{aligned}
V_t &= e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} \left[(S_T^{(1)} - S_T^{(2)})^+ | \mathcal{F}_t \right] \\
&= \frac{\beta_t}{\beta_T} \mathbb{E}^{\mathbb{Q}} \left[S_T^{(2)} \left(\frac{S_T^{(1)}}{S_T^{(2)}} - 1 \right)^+ | \mathcal{F}_t \right] \\
&= \beta_t \mathbb{E}^{\mathbb{Q}} \left[\left(\frac{S_T^{(1)}}{S_T^{(2)}} - 1 \right)^+ \frac{S_T^{(2)}}{\beta_T} | \mathcal{F}_t \right] \\
&= S_t^{(2)} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[\left(\frac{S_T^{(1)}}{S_T^{(2)}} - 1 \right)^+ | \mathcal{F}_t \right]
\end{aligned} \tag{90}$$

Using $S_t^{(2)}$ as numéraire, we have managed to go back to the Black-Scholes-Merton formula. Indeed, the option to exchange $S_t^{(2)}$ for $S_t^{(1)}$ is found to be an European call option with spot price $S_t^{(1)}/S_t^{(2)}$ and strike equal to unity.

We now have two possibilities to price the exchange option: continue solving this expectation with Girsanov theorem, or simply using the BSM European call option pricing formula whilst adapting some constants.

Pricing Exchange option with European call pricing formula

In this second case, one must: consider a null risk-free rate ([Margrabe, 1978](#)) and the volatility of the spot price $S_t^{(1)}/S_t^{(2)}$ is given by $\sigma = \sqrt{(\sigma^{(1)})^2 + (\sigma^{(2)})^2 - 2\sigma^{(1)}\sigma^{(2)}\rho}$. Note that the resulting EU BSM option price will be computed in units of $S_t^{(2)}$ given the change of numéraire.

Mathematically speaking, we thus have ([Margrabe, 1978](#)):

$$\begin{aligned}
V_t &= S_t^{(2)} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[\left(\frac{S_T^{(1)}}{S_T^{(2)}} - 1 \right)^+ | \mathcal{F}_t \right] \\
V_t &= S_t^{(2)} \underbrace{\left[\frac{S_t^{(1)}}{S_t^{(2)}} \Phi(d_1) - 1 e^{-0*(T-t)} \Phi(d_2) \right]}_{V_t \text{ in units of } S_t^{(2)} \text{ from BSM EU call}} \\
V_t &= S_t^{(1)} \Phi(d_1) - S_t^{(2)} \Phi(d_2)
\end{aligned} \tag{91}$$

Where $d_1 = \frac{\ln \frac{S_t^{(1)}}{S_t^{(2)}} + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T}}$, and $d_2 = d_1 - \sigma\sqrt{T-t}$.

Pricing Exchange option with Girsanov theorem

The full proof can be found below and was inspired by: (Margrabe, 1978), (Shreve, 2004b), (Bodart, Henry, Kinart, & Vanderveken, 2018), (Vrins, 2020).

Starting from the expectation using ${}_t^{(2)}$ as numéraire,:

$$V_t = S_t^{(2)} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[\left(\frac{S_T^{(1)}}{S_T^{(2)}} - 1 \right)^+ | \mathcal{F}_t \right] \quad (92)$$

Creating $Y_t = \frac{S_t^{(1)}}{S_t^{(2)}}$, we define an indicator function telling at maturity if $S_T^{(1)} > S_T^{(2)}$:

$$\mathbb{1} = \begin{cases} 1, & \text{if } Y_T > 1 \\ 0, & \text{otherwise} \end{cases}$$

The expectation thus becomes:

$$\begin{aligned} V_t &= S_t^{(2)} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[(Y_T - 1) \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t) \right] \\ &= S_t^{(2)} \left(\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[Y_T \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t) \right] - \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t) \right] \right) \end{aligned} \quad (93)$$

The proof will now be structured as such: deriving first the right expectation, then the left and finally putting them together.

$$\underline{\textbf{Right expectation: } \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} \left[\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t) \right]}$$

We first need to compute the dynamics of the process Y_t under $\mathbb{Q}^{S_t^{(2)}}$.

To do so, let us start with reminding Ito's lemma in 2 dimensions where X and Y are two continuous stochastic processes,

$$\begin{aligned} df(t, X_t, Y_t) &= f_t(t, X_t, Y_t)dt + f_x(t, X_t, Y_t)dX_t + f_y(t, X_t, Y_t)dY_t \\ &\quad + \frac{1}{2} [f_{xx}(t, X_t, Y_t)(dX_t)^2 + 2f_{xy}(t, X_t, Y_t)dX_t dY_t + f_{yy}(t, X_t, Y_t)(dY_t)^2] \end{aligned} \quad (94)$$

For conciseness, let us write $f_t = f_t(t, X_t, Y_t)$, $f_x = f_x(t, X_t, Y_t)$, $f_{xx} = f_{xx}(t, X_t, Y_t)$, $f_y = f_y(t, X_t, Y_t)$, $f_{yy} = f_{yy}(t, X_t, Y_t)$, $f_{xy} = f_{xy}(t, X_t, Y_t)$.

Applying it to $Y_t = f(t, S_t^{(1)}, S_t^{(2)})$,

$$\begin{aligned}
f_t &= 0 \\
f_x &= \frac{1}{S_t^{(2)}} \\
f_{xx} &= 0 \\
f_y &= -\frac{S_t^{(1)}}{(S_t^{(2)})^2} \\
f_{yy} &= \frac{2S_t^{(1)}}{(S_t^{(2)})^3} \\
f_{xy} &= -\frac{1}{(S_t^{(2)})^2}
\end{aligned} \tag{95}$$

Ito thus gives:

$$\begin{aligned}
dY_t &= \frac{1}{S_t^{(2)}} dS_t^{(1)} - \frac{S_t^{(1)}}{(S_t^{(2)})^2} dS_t^{(2)} + \frac{1}{2} \left[-2 \frac{1}{(S_t^{(2)})^2} dS_t^{(1)} dS_t^{(2)} + \frac{2S_t^{(1)}}{(S_t^{(2)})^3} (dS_t^{(2)})^2 \right] \\
&= \left[\frac{-2dS_t^{(1)} dS_t^{(2)} S_t^{(2)} + 2S_t^{(1)} (dS_t^{(2)})^2}{(S_t^{(2)})^3} \right]
\end{aligned} \tag{96}$$

Given that $(dt)^2 = dt dW_t = 0$ and $(dW_t)^2 = dt$,

$$\begin{aligned}
&= \frac{1}{2} \left[\frac{-2\sigma^{(1)}\sigma^{(2)} S_t^{(1)} \cancel{S_t^{(2)}} dW_t^{(1)} dW_t^{(2)} \cancel{S_t^{(2)}} + 2S_t^{(1)} (\sigma^{(2)})^2 \cancel{(S_t^{(2)})^2} dt}{(S_t^{(2)})^3} \right] \\
&= \frac{1}{2} \left[\frac{S_t^{(1)}}{S_t^{(2)}} (-2\sigma^{(1)}\sigma^{(2)} \rho dt + 2(\sigma^{(2)})^2 dt) \right] \\
&= \frac{S_t^{(1)}}{S_t^{(2)}} \left[\frac{1}{S_t^{(1)}} dS_t^{(1)} - \frac{1}{S_t^{(2)}} dS_t^{(2)} + \frac{1}{2} [-2\sigma^{(1)}\sigma^{(2)} \rho dt + 2(\sigma^{(2)})^2 dt] \right] \\
&= Y_t \left[\mu^{(1)} dt + \sigma^{(1)} dW_t^{(1)} - \mu^{(2)} dt - \sigma^{(2)} dW_t^{(2)} - \sigma^{(1)}\sigma^{(2)} \rho dt + (\sigma^{(2)})^2 dt \right] \\
&= Y_t \left[(\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)} \rho + (\sigma^{(2)})^2) dt + \sigma^{(1)} dW_t^{(1)} - \sigma^{(2)} dW_t^{(2)} \right]
\end{aligned} \tag{97}$$

Creating a new Brownian motion $\sigma dW_t = \sigma^{(1)} dW_t^{(1)} - \sigma^{(2)} dW_t^{(2)}$ where $\sigma^2 = (\sigma^{(1)})^2 + (\sigma^{(2)})^2 - 2\sigma^{(1)}\sigma^{(2)}\rho$, it is found that:

$$dY_t = Y_t [(\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)}\rho + (\sigma^{(2)})^2)dt + \sigma dW_t] \tag{98}$$

Where W_t is a $\mathbb{P}$ -Brownian motion.

So far, we have derived the dynamics of Y_t under $\mathbb{P}$, but not yet under $\mathbb{Q}^{S_t^{(2)}}$. As we would like to get the arbitrage-free option price, knowing the dynamics under $\mathbb{Q}^{S_t^{(2)}}$ (which is equivalent to the risk-neutral probability thanks to the change of numéraire!) is primordial.

In other words, we want the process Y_t to be driftless such that it becomes a martingale. We obviously currently see that is not the case. We can thus use the Girsanov theorem, which tells how the dynamics of stochastic processes change when changing probability measures.

Starting with the previously created W_t , a $\mathbb{P}$ -Brownian motion, the theorem states that using the Radon-Nikodym derivative ζ_t defined as

$$\zeta_t = \exp\left\{-\int_0^t \theta_s dW_s - \frac{1}{2} \int_0^t \theta_s^2 ds\right\} \quad (99)$$

Which respects $\mathbb{P}(\zeta_t > 0) = 1$ and $\mathbb{E}[\zeta_t] = 1$. Then, the process:

$$\tilde{W}_t := W_t + \int_0^t \theta_s ds \quad (100)$$

Is a $\mathbb{Q}^{S_t^{(2)}}$ -Brownian motion for $0 \leq t \leq T$. Assuming θ deterministic, we now simply have to choose it such that the process Y_t is driftless, such that it is a martingale.

Mathematically speaking, that is translated by

$$\begin{aligned} d\tilde{W}_t &= dW_t + \theta dt \\ &= dW_t + \frac{\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)}\rho + (\sigma^{(2)})^2}{\sigma} dt \end{aligned} \quad (101)$$

Replacing dW_t by $d\tilde{W}_t$ in the Y_t process, we have that:

$$\begin{aligned} dY_t &= Y_t \left[(\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)}\rho + (\sigma^{(2)})^2) dt + \sigma dW_t \right] \\ &= Y_t \left[\cancel{(\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)}\rho + (\sigma^{(2)})^2) dt} + \sigma \left[d\tilde{W}_t - \cancel{\frac{\mu^{(1)} - \mu^{(2)} - \sigma^{(1)}\sigma^{(2)}\rho + (\sigma^{(2)})^2}{\sigma} dt} \right] \right] \\ &= Y_t \sigma d\tilde{W}_t \end{aligned} \quad (102)$$

We have finished deriving the dynamics of Y_t under $\mathbb{Q}^{S_t^{(2)}}$ and showing that it is a martingale.

Now we need to compute the solution of Y_t under $\mathbb{Q}^{S_t^{(2)}}$.

This is quite fast: using 1D Ito's lemma exactly as in the earlier appendices with $\ln(x)$ as substitute function, we find that the solution to the SDE $dY_t = Y_t \sigma d\tilde{W}_t$ is given by:

$$Y_t = Y_0 e^{-\frac{\sigma^2}{2}t + \sigma \tilde{W}_t} \quad (103)$$

Let us now go back to the right expectation, given by

$$\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] \quad (104)$$

This became a trivial problem once the dynamics and solution are known:

$$\begin{aligned} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] &= \mathbb{Q}^{S_t^{(2)}} (Y_T > 1) \\ &= \mathbb{Q}^{S_t^{(2)}} \left(Y_t e^{-\frac{\sigma^2}{2}(T-t) + \sigma(\tilde{W}_T - \tilde{W}_t)} > 1 \right) \\ &= \mathbb{Q}^{S_t^{(2)}} \left(Z > \frac{-\ln(Y_t) + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}} \right) \end{aligned} \quad (105)$$

We define $-d_2 = \frac{-\ln(Y_t) + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}$, and from the Normal distribution, it is known that $P(Z > -x) = P(Z < x)$. Therefore,

$$\begin{aligned} &= \mathbb{Q}^{S_t^{(2)}} (Z < d_2) \\ &= \Phi(d_2) \end{aligned} \quad (106)$$

The solution to the right expectation is thus $\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] = \Phi(d_2)$, with

$$d_2 = \frac{\ln(Y_t) - \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}$$

Left expectation: $\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [Y_T \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)]$

For the left expectation, the goal will now be to obtain a simple expectation similar to the right one, i.e., with only the indicator function.

For that, we will use the Radon-Nikodym and its properties. Working under $\mathbb{Q}^{S_t^{(2)}}$, we define RND ζ_t as:

$$\zeta_t = \exp\left\{-\int_0^t \theta_s d\tilde{W}_s - \frac{1}{2} \int_0^t \theta_s^2 ds\right\} \quad (107)$$

Which respects $\mathbb{Q}^{S_t^{(2)}}(\zeta_t > 0) = 1$ and $\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}}[\zeta_t] = 1$. With it, we can create the probability measure $\bar{\mathbb{Q}}$ equivalent to $\mathbb{Q}^{S_t^{(2)}}$!

We now observe that for a deterministic $\theta = -\sigma$, we find that $\zeta_t = \frac{Y_t}{Y_0}$.

$$\begin{aligned} \zeta_t &= e^{-\int_0^t (-\sigma) d\tilde{W}_s - \frac{1}{2} \int_0^t (-\sigma)^2 ds} \\ &= e^{\sigma \tilde{W}_t - \frac{\sigma^2}{2} t} = \frac{Y_t}{Y_0} \end{aligned} \quad (108)$$

This allows us to use a RND property to change the probability measure of the left expectation to the newly created equivalent measure $\bar{\mathbb{Q}}$.

$$\begin{aligned} \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\zeta_T Y_t \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] &= Y_t \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\zeta_T \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] \\ &= Y_t \mathbb{E}^{\bar{\mathbb{Q}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] \end{aligned} \quad (109)$$

We are left to see how the dynamics of Y_t changed from $\mathbb{Q}^{S_t^{(2)}}$ to $\bar{\mathbb{Q}}$. Once again we use Girsanov theorem thanks to the RND ζ_t given in equation (108) and define the process:

$$\bar{W}_t := \tilde{W}_t - \sigma t \quad (110)$$

Reminding the dynamics of Y_t under $\mathbb{Q}^{S_t^{(2)}}$ and plugging $\bar{W}_t$ in it,

$$\begin{aligned} dY_t &= Y_t \sigma d\tilde{W}_t \\ &= Y_t \sigma (d\bar{W}_t + \sigma dt) \\ &= Y_t \sigma^2 dt + Y_t \sigma d\bar{W}_t \end{aligned} \quad (111)$$

We now only have to solve the stochastic equation dY_t under $\bar{W}_t$. Using 1D Ito's lemma exactly as in the earlier appendices with $\ln(x)$ as substitute function, we find that the solution

to the SDE $dY_t = Y_t \sigma^2 dt + Y_t \sigma d\bar{W}_t$ is given by:

$$Y_t = Y_0 e^{\frac{\sigma^2}{2}t + \sigma \bar{W}_t} \quad (112)$$

Going back to the left expectation under $\bar{W}_t$ given in equation (109),

$$\begin{aligned} Y_t \mathbb{E}^{\bar{\mathbb{Q}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] &= Y_t \bar{\mathbb{Q}}(Y_T > 1) \\ &= Y_t \bar{\mathbb{Q}} \left(Y_t e^{\frac{\sigma^2}{2}(T-t) + \sigma(\bar{W}_T - \bar{W}_t)} > 1 \right) \\ &= Y_t \bar{\mathbb{Q}} \left(Z > \frac{-\ln(Y_t) - \frac{\sigma^2}{2}(T-t)}{\sigma \sqrt{T-t}} \right) \end{aligned} \quad (113)$$

We define $-d_1 = \frac{-\ln(Y_t) - \frac{\sigma^2}{2}(T-t)}{\sigma \sqrt{T-t}}$, and from the Normal distribution, it is known that $P(Z > -x) = P(Z < x)$. Therefore,

$$\begin{aligned} &= Y_t \bar{\mathbb{Q}}(Z < d_1) \\ &= Y_t \Phi(d_1) \end{aligned} \quad (114)$$

The solution to the left expectation is thus $\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [Y_T \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] = \Phi(d_1)$, with $d_1 = \frac{\ln(Y_t) + \frac{\sigma^2}{2}(T-t)}{\sigma \sqrt{T-t}}$

Back to the risk-neutral pricing formula

After computing both left and right expectations, we plug them together in the exchange option pricing formula (93):

$$\begin{aligned} V_t &= S_t^{(2)} \left(\mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [Y_T \mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] - \mathbb{E}^{\mathbb{Q}^{S_t^{(2)}}} [\mathbb{1}_{\{Y_T > 1\}} | \mathcal{F}(t)] \right) \\ &= S_t^{(2)} [Y_t \Phi(d_1) - \Phi(d_2)] \\ &= S_t^{(1)} \Phi(d_1) - S_t^{(2)} \Phi(d_2) \end{aligned} \quad (115)$$

Which is the exchange option pricing formula found through risk-neutral expectation with the Girsanov theorem, the same as Margrabe found in 1978.

[\[Return to 2.4 Risk-neutral pricing formulas - Exchange options\]](#)

8.10 BSM Exchange option replication strategy

We infer the dynamics of the option price by applying Ito's lemma to the BSM PDE V_t given in equation (12). Applying two variable Ito's lemma to $V_t = f(t, S_t^{(1)}, S_t^{(2)})$:

$$\begin{aligned} df(t, S_t^{(1)}, S_t^{(2)}) = & f_t(t, S_t^{(1)}, S_t^{(2)})dt + f_x(t, S_t^{(1)}, S_t^{(2)})dX_t + f_y(t, S_t^{(1)}, S_t^{(2)})dY_t \\ & + \frac{1}{2} \left[f_{xx}(t, S_t^{(1)}, S_t^{(2)})(dX_t)^2 + 2f_{xy}(t, S_t^{(1)}, S_t^{(2)})dX_t dY_t + f_{yy}(t, S_t^{(1)}, S_t^{(2)})(dY_t)^2 \right] \end{aligned} \quad (116)$$

Shortening the notation and applying to our case,

$$\begin{aligned} dV_t = & f_t dt + f_x dS_t^{(1)} + f_y dS_t^{(2)} + \frac{1}{2} \left[f_{xx}(\sigma^{(1)})^2 (S_t^{(1)})^2 dt + 2f_{xy}\sigma^{(1)}\sigma^{(2)}S_t^{(1)}S_t^{(2)} \underbrace{dW_t^{(1)}dW_t^{(2)}}_{\rho dt} + f_{yy}(\sigma^{(2)})^2 (S_t^{(2)})^2 dt \right] \\ = & \left[f_t + \frac{1}{2} \left(f_{xx}(\sigma^{(1)})^2 (S_t^{(1)})^2 + 2f_{xy}\sigma^{(1)}\sigma^{(2)}S_t^{(1)}S_t^{(2)}\rho + f_{yy}(\sigma^{(2)})^2 (S_t^{(2)})^2 \right) \right] dt + f_x dS_t^{(1)} + f_y dS_t^{(2)} \end{aligned} \quad (117)$$

Let us continue by creating a portfolio Π_t that will replicate at all $t \in [0; T]$ the derivative price. At time t , invest in $\Delta_t^{(1)}$ and $\Delta_t^{(2)}$ shares of each stock. The portfolio will therefore be composed of the value of fraction of shares and the cash reminder.

$$\Pi_t = \underbrace{\Pi_t - \Delta_t^{(1)}S_t^{(1)} - \Delta_t^{(2)}S_t^{(2)}}_{\text{cash account}} + \underbrace{\Delta_t^{(1)}S_t^{(1)} + \Delta_t^{(2)}S_t^{(2)}}_{\text{equity account}} \quad (118)$$

Given the self-financing assumption of the portfolio,

$$d\Pi_t = \underbrace{r(\Pi_t - \Delta_t^{(1)}S_t^{(1)} - \Delta_t^{(2)}S_t^{(2)})dt}_{\text{cash grows at risk-free rate}} + \underbrace{\Delta_t^{(1)}dS_t^{(1)} + \Delta_t^{(2)}dS_t^{(2)}}_{\text{equity changes are only due to stock changes}} \quad (119)$$

Let us now equate (119) to (117):

$$\begin{aligned} d\Pi_t &= dV_t \\ r(\Pi_t - \Delta_t^{(1)}S_t^{(1)} - \Delta_t^{(2)}S_t^{(2)})dt + \Delta_t^{(1)}dS_t^{(1)} + \Delta_t^{(2)}dS_t^{(2)} &= \\ \left[f_t + \frac{1}{2} \left(f_{xx}(\sigma^{(1)})^2 (S_t^{(1)})^2 + 2f_{xy}\sigma^{(1)}\sigma^{(2)}S_t^{(1)}S_t^{(2)}\rho + f_{yy}(\sigma^{(2)})^2 (S_t^{(2)})^2 \right) \right] dt &+ f_x dS_t^{(1)} + f_y dS_t^{(2)} \end{aligned} \quad (120)$$

Observe that if:

$$\left\{ \begin{array}{l} r(\Pi_t - \Delta_t^{(2)} S_t^{(2)}) = f_t + \frac{1}{2} \left(f_{xx} (\sigma^{(1)})^2 (S_t^{(1)})^2 + 2f_{xy} \sigma^{(1)} \sigma^{(2)} S_t^{(1)} S_t^{(2)} \rho + f_{yy} (\sigma^{(2)})^2 (S_t^{(2)})^2 \right) \\ \Delta_t^{(1)} = f_x \\ \Delta_t^{(2)} = f_y \end{array} \right. \quad (121)$$

Then $\Pi_t = V_t$. In other words, as long as the strategy holds both $\Delta_t^{(1)} = \Phi(d_1)$ and $\Delta_t^{(2)} = -\Phi(d_2)$ shares of stocks 1 and 2 at all times t and re-balance the portfolio accordingly, it perfectly replicates the BSM pricing formula. This is thus called a delta-hedge portfolio, as it hedges away the randomness embedded in $S_t^{(1)}$ and $S_t^{(2)}$.

We have proved that the price of a derivative is the cost of setting up a self-financing trading strategy replicating the derivative's payoff whatever happens in the future (Vrins, 2020).

Note that the portfolio re-balancing equation can be simplified. Indeed, for the replication of exchange options, the cash is in reality always perfectly allocated between both stocks. There is thus no cash in the cash account.

The re-balancing equation of the portfolio is thus simplified into:

$$d\Pi_t = \Delta_t^{(1)} dS_t^{(1)} + \Delta_t^{(2)} dS_t^{(2)} \quad (122)$$

And the above results are obviously still valid.

[\[Return to 2.5 Replication strategy - Exchange options\]](#)

8.11 Permissive MIT license used in the applications

This license template was taken from the official open-source repository of internet licenses (*The MIT License*, [n.d.](#)). It must be and can be found in the source folder of all web applications.

MIT License

Copyright (c) 2020 Michel Vanderhulst

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[\[Return to subsection 3.3 App legal aspects\]](#)

8.12 Example of generated Excel file in European option web app

	A	B	C	D	E	F
1		0	1	2	3	4
2	1	100	128.1723	164.2813	210.5631	269.8835
3	2		90.6463	116.1834	148.9149	190.8677
4	3			82.16751	105.316	134.9859
5	4				74.48181	95.46503
6	5					67.515

Figure 27: Example of Excel file for European option web app

[\[Return to subsection 4.2.1 Example 1: European option CRR web app\]](#)

8.13 Example of generated Excel file in Margrabe option web app

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Time (in dt)	Stock price 1	Stock price 2	Option intrc. value	Option price	Cash before	Equity 1 before	Equity 2 before	Portfolio before	Delta 1	Delta 2	Cash after	Equity 1 after	Equity 2 after	Portfolio after	Replication strategy error	
2	0	100	100	0	17.747961	17.747961	0	0	17.747961	0.58874	-0.41126	0	58.87398	-41.12602	17.747961	0	
3	0.01	104.216068	101.355002	2.861066	19.681798	0	61.356148	-41.683278	19.67287	0.612609	-0.435715	0	63.843669	-44.161871	19.681798	0	
4	0.02	102.704908	101.953861	0.751048	18.488198	0	62.917919	-44.422803	18.495117	0.594862	-0.417905	0	61.09526	-42.607062	18.488198	0	
5	0.03	101.427384	101.496582	0	17.896794	0	60.335309	-42.415963	17.919347	0.587771	-0.411041	0	59.616052	-41.719257	17.896794	0	
6	0.04	98.81075	101.843689	0	16.229193	0	58.07807	-41.861933	16.216138	0.561683	-0.385602	0	55.500347	-39.271154	16.229193	0	
7	0.05	101.041274	101.321132	0	17.691401	0	56.753196	-39.069655	17.683541	0.585696	-0.409471	0	59.179443	-41.488041	17.691401	0	
8	0.06	95.457214	100.004542	0	15.01722	0	55.908882	-40.948936	14.959946	0.546861	-0.37183	0	52.201863	-37.184643	15.01722	0	
9	0.07	99.781812	100.573691	0	17.210551	0	54.566819	-37.39627	17.170549	0.580921	-0.405223	0	57.965364	-40.754813	17.210551	0	
10	0.08	97.96823	100.057247	0	16.348891	0	56.911817	-40.545538	16.366279	0.569125	-0.393847	0	55.75616	-39.407269	16.348891	0	
11	0.09	98.820663	99.620595	0	16.988347	0	56.241301	-39.235295	17.006005	0.580509	-0.405317	0	57.366239	-40.377892	16.988347	0	
12	0.1	98.274046	98.805776	0	16.976675	0	57.048923	-40.047632	17.001291	0.582726	-0.407772	0	57.266864	-40.290189	16.976675	0	
13	0.11	102.00279	98.610153	3.392637	19.271849	0	59.439701	-40.210419	19.229282	0.616907	-0.442696	0	62.926216	-43.654367	19.271849	0	
14	0.12	96.948911	98.780765	0	16.172751	0	59.808444	-43.729897	16.078547	0.570625	-0.396319	0	55.32146	-39.148709	16.172751	0	
15	0.13	96.236742	96.771016	0	16.545751	0	54.915078	-38.352207	16.562871	0.582199	-0.408006	0	56.028946	-39.483194	16.545751	0	
16	0.14	95.382713	96.883376	0	15.982556	0	55.531731	-39.529038	16.002693	0.573094	-0.39925	0	54.663279	-38.680723	15.982556	0	
17	0.15	98.192427	96.892662	1.299765	17.600397	0	56.273509	-38.68443	17.589079	0.598632	-0.425013	0	58.781086	-41.180689	17.600397	0	
18	0.16	95.594885	97.144252	0	15.950869	0	57.226115	-41.287618	15.938497	0.572377	-0.399051	0	54.716353	-38.765485	15.950869	0	
19	0.17	95.249151	97.469861	0	15.600517	0	54.518463	-38.895419	15.623044	0.565969	-0.393019	0	53.908028	-38.307511	15.600517	0	
20	0.18	93.245625	94.567698	0	15.58651	0	52.774096	-37.166906	15.60719	0.573871	-0.401029	0	53.510945	-37.924435	15.58651	0	
21	0.19	93.40828	95.008824	0	15.479156	0	53.604288	-38.10134	15.502948	0.571105	-0.39856	0	53.345907	-37.866751	15.479156	0	
22	0.2	94.84442	95.369764	0	16.136632	0	54.166094	-38.010608	16.155486	0.581261	-0.408859	0	55.129377	-38.992745	16.136632	0	
23	0.21	92.333765	94.975436	0	14.836001	0	53.670031	-38.831521	14.83851	0.560649	-0.388846	0	51.766866	-36.930865	14.836001	0	
24	0.22	95.079699	94.777459	0.302241	16.469884	0	53.306373	-36.853882	16.452491	0.588805	-0.416908	0	55.983396	-39.513511	16.469884	0	
25	0.23	97.313989	95.710323	1.603666	17.379441	0	57.298956	-39.90243	17.396526	0.600689	-0.42917	0	58.455409	-41.075969	17.379441	0	
26	0.24	98.611965	95.377585	3.23438	18.289103	0	59.235089	-40.933168	18.301921	0.615432	-0.444547	0	60.688913	-42.39981	18.289103	0	
27	0.25	100.927406	97.222119	3.705287	18.869287	0	62.113908	-43.219792	18.894116	0.618929	-0.448433	0	62.466879	-43.597591	18.869287	0	
28	0.26	99.28513	97.4443	1.840829	17.743155	0	61.450427	-43.697225	17.753202	0.602249	-0.431541	0	59.794326	-42.051171	17.743155	0	
29	0.27	99.048641	99.310951	0	16.789848	0	59.651901	-42.856706	16.795195	0.582909	-0.412306	0	57.736385	-40.946537	16.789848	0	

Figure 28: Example of Excel file for Margrabe web app

[\[Return to subsubsection 4.2.2 Example 2: Exchange option BSM web app\]](#)

8.14 Example of error handling in European option CRR web app



Figure 29: European option web app negative stock price and tree periods error handling

In the input sub-tab, we observe in red that the software indicates to the user his problematic input.

[\[Return to subsubsection 4.2.1 Example 1: European option CRR web app\]](#)

