Stability of the Numerical Schemes used for Pricing Green Bonds

by

Patrick MacDonald

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday, September 3, 2024, at 15:30 PM.

Student number:	5875378
Project duration:	November 13, 2023 - September 3, 2024
Thesis committee:	Prof. dr. ir. C. Vuik
	Dr. ir. mr. V.N.S.R. (Vandana) Dwarka
	Prof. dr. ir. H.X. Lin

An electronic version of this thesis is available at
`http://repository.tudelft.nl/`

Acknowledgements

With these few words, I would like to thank my supervisors, Professor Cornelis Vuik and Assistant Professor V.N.S.R. Dwarka, for their patience, guidance, and support during this project. The Master's program in Applied Mathematics is very challenging, but with their help, I completed this master's project without major problems. For this, I am very grateful. My questions were always answered extensively and quickly and our meetings (both online and in person) helped me to develop this thesis. Moreover, I want to thank Professor Hai Xiang Lin for being part of my thesis committee.

Last but definitely not least, I want to thank my family, who were always there in case I needed them.

I hope you have a good time reading the thesis,

Patrick MacDonald,
Delft, August 2024

Abstract

This master's thesis provides information on the subject of option pricing theory. Moreover, this topic is linked with sustainable Finance, which is essential in the battle against climate change. Green financing is a growing phenomenon, and a green bond is only a relatively new example of a green financial derivative. A three-dimensional PDE is at hand to determine the price of a green bond (also called a coupon value), but this PDE can only be solved numerically. This master thesis aims to determine the stability of certain numerical schemes that can be used to find a solution to the PDE. We will use both forward and backward differences to derive the numerical schemes. After the derivation process, a Von Neumann analysis is performed to conclude whether or not the schemes are stable. The research reveals that most of the numerical schemes are not stable. Amongst others, this is caused by the positive value of the interest rate r, the value of the carbon price, and the absence of damping factors. In some numerical schemes, the amplification factor is larger than one, but not by much. In other schemes, we can reduce the amplification factor to increase the stability. This means that the numerical schemes can still be used to find a reasonable value for a green bond.

Keywords: Option pricing theory; Black-Scholes; heat equation; green finance; finite difference discretization; Forward Euler; Backward Euler; amplification factor; Von Neumann stability; Conjugate Gradient; Bi-CGSTAB; GMRES.

Contents

List of Figures

List of Tables

List of Abbreviations

PDE	Partial Differential Equation
SDE	Stochastic Differential Equation
ICMA	International Capital Market Association
EUGBS	European Green Bond Standard
bps	basis points
BS	Black-Scholes
BM	Brownian Motion
SPD	Symmetric and Positive Definite
FE	Forward Euler
BE	Backward Euler
CN	Crank-Nicolson
FD	Finite Difference
CG	Conjugate Gradient
Bi-CGSTAB	Bi-Conjugate Gradient Stabilized
GMRES	General Minimal Residual method
FTCS	Forward difference in Time, Central difference in Space
BTCS	Backward difference in Time, Central difference in Space

1 Option Pricing and Green Finance

1.1 Option Theory

According to [1], an asset describes any financial object whose value is known at present but is liable to change in the future. In this chapter, we will come across multiple options with an underlying asset. For now, we can consider an option to be a contract between two parties. We will introduce different types of options and elaborate on how an option's price can be determined. This price must be fair for both the option seller (who is also called the writer) and the option buyer (who is also called the holder).

1.1.1 Bonds

The most common options that occur in finance are bonds. A bond is a contract between two parties, an investor and a borrower. For the borrowing party, the money can be used to finance important projects that might not have been possible to realize without the money. Of course, the money has to be paid back to the investor after a pre-specified amount of time. The investor takes a risk because the borrower might not be able to pay the money back. Therefore, the money must be paid back with an additional feed to compensate for this risk. This fee is called the interest. The interest is determined before the two parties sign the contract. Sometimes, the interest is paid all at once on the final day before the money is paid back, but it could also be the case that the interest is paid in installments (e.g., annually, semi-annually, monthly, etc.). Since both the time when a lender gets his money back and the interest payments are known in advance, bonds are referred to as (predictable) fixed-income securities.

The interest rate, usually denoted by r, depends on the risk the investor takes. There are tools to measure the solvability of companies willing to borrow money. Roughly said: a company with a low solvency ratio (indicating a bad financial position) has to pay more interest than another company that is, from a financial point of view, perfectly healthy. Other factors that influence the value of r are the amount of money borrowed and how quickly the borrower plans to repay the money.

The most elemental version of a bond is the so-called zero-coupon bond. The definition is derived from [2].

Definition 1 (**Zero-coupon bond**). *A zero-coupon bond with a value at time t with maturity time T, denoted by $B(t, T)$, is a financial instrument that can be bought at time $t = 0$ for a price of $B(0, T)$, and pays one unit of currency (for example, euro, dollar, etc.) at maturity time T, i.e., $B(T, T) = 1$.*

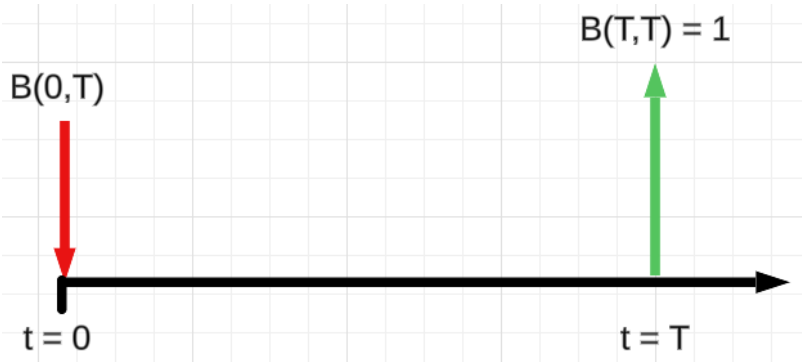


Figure 1: Payments for a zero-coupon bond with maturity time T.

Assume that the interest rate r is constant and that we deal with continuously compounded interest. Then, the value of the bond at time $t \in [0,T]$, denoted by $B(t,T)$, has to satisfy the equation

$$e^{r(T-t)}B(t,T) = B(T,T) = 1.$$

From here it follows that the value of the bond is given by

$$\forall t \in [0,T] : B(t,T) = e^{-r(T-t)}.$$

If the interest rate is assumed to vary smoothly over time, i.e., $r : [0,T] \to \mathbb{R}$ is a differentiable function of time, the value of the bond is given by

$$B(t,T) = e^{-\int_t^T r(s)ds}.$$

The next bond that we will consider is a coupon bond. For such a bond, intermediate payoffs can occur before one unit of currency is paid at time T. Determining the price of such an option is much more difficult. Normally, this value can only be computed numerically instead of analytically.

Definition 2 (**Coupon bond**). *A coupon bond with a value at time t with maturity time T, denoted by $B(t,T)$, is a financial instrument that can be bought at time $t = 0$ for a price of $B(0,T)$. At a finite number of times t_j for $1 \leq j \leq n$, the buyer receives a payment of c_j. At the maturity time, the coupon buyer gets an amount of $B(T,T) = 1 + c_n$.*

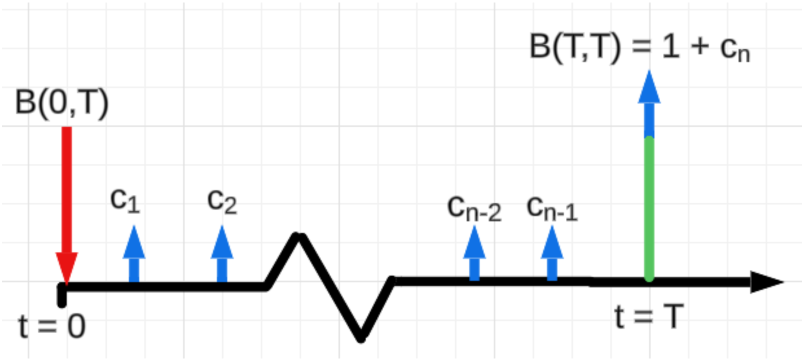


Figure 2: Payments for a coupon bond with maturity time T.

1.1.2 Call and put options

In addition to bonds, call and put options are also types of options that occur regularly in practice. One reason to buy or sell such an option is to hedge risk. These types of options are often called **derivatives** since the value depends on the value of an underlying asset. The next definition is cited from [2].

Definition 3 (**European call option**). *A European call option gives an option holder the right, but not the obligation, to purchase an asset at a pre-specified time in the future, denoted by $t = T$ (the maturity time), for a prescribed amount of money. This amount is called the strike price and is denoted by K.*

The writer of a European call option hopes the asset price will fall after selling the option, whereas the holder of such an op option hopes the asset price will rise after buying the option. At maturity time T, the contract holder either

- makes a profit if the price of the asset exceeds the strike price, or
- does not gain or lose money if the price is lower than the strike price.

In both cases, the holder of the contract does not lose money. On the other hand, the person who wrote out the option does not gain any money in both cases. To compensate for this dishonesty, the contract holder is supposed to pay the writer of the contract an amount of money at time $t = 0$. This amount is called the **value** of the call option. For any $t \in [0, T]$, we denote the value of a call option by $V_{\text{call}}(t, S)$.

The opposite of a European call option is a European put option. The definition is also cited from [2].

Definition 4 (**European put option**). *A European put option gives an option holder the right, but not the obligation, to sell an asset at a pre-specified time in the future, denoted by $t = T$ (the maturity time), for a prescribed strike price K.*

The writer of a European put option hopes the asset price will rise after selling the option, whereas the holder hopes the asset price will fall after buying the option. At maturity time T, the contract holder either

- makes a profit if the price of the asset is lower than the strike price, or
- does not gain or lose money if the price exceeds the strike price.

Again, the contract holder is supposed to pay the writer of the contract an amount of money at time $t = 0$, which is called the value of the put option. This value is denoted by $V_{\text{call}}(t, S)$. In general, determining the value of a European call or put option is not easy. Later in this chapter, we will show a formula that can be used to determine the value.

If the option holder decides to buy or sell the asset at the maturity time, we say that the holder has **exercised** the contract. The following example demonstrates the difference between a European call and a European put option.

Example 1. *Suppose person X and person Y are interested in a stock S. At time $t = 0$, its value is equal to $S_0 =$ €10,-. Person X decides to buy a European call option on this stock with strike price $K =$ €9,- and maturity time $T = 1$, corresponding to one week. Person Y on the other hand chooses to buy both the stock for €10,- and a European put option on this stock for the same values of K and T.*

After one week, the value of the stock is increased: $S_1 =$ €12,-. Both person X and Y now have to make a decision.

- *Person X has the right to buy the stock for the strike price of €9,-. Since the value of the stock is now higher, person X of course decides to do this (he decides to exercise the option). By buying the stock for €9,- and selling it directly on the stock market for its current price, which is €12,-, person X makes a profit of €3,-. This amount is called the **payoff** of the contract.*
- *Person Y can also decide whether or not to exercise his option. As a reminder, he has the right to sell the stock for the strike price of €9,-. Since the value of the stock is now higher, person Y makes a rational decision by not exercising the option, i.e., his payoff is zero. If he exercised the option, he would sell his stock for only €9,-, whereas on the stock market, he could get €12,-.*

If someone buys or sells a European option at time $t = 0$, this person knows that his only chance to exercise the contract is at the pre-specified maturity date $t = T$. This works differently for a **Bermudan** call or put option. For this kind of option, a holder can exercise the contract at a finite number of pre-specified dates $t_1, t_2, \ldots, t_n \in [0, T]$ (for example, weekly or monthly). If the holder of a call or put option can exercise the contract at any time $t \in [0, T]$, then we speak of an **American** option.

1.1.3 Payoff diagram of a call and put option

For a call option, it is not wise to exercise the contract if the asset value at the maturity time is smaller than or equal to the strike price, i.e., $S_T \leq K$. If someone decides to exercise the call option under this condition, this person would buy the asset for a price of K, while it is worth less on the market. Therefore, the call option is not exercised and the payoff is zero. On the other hand, if $S_T > K$, the contract holder should exercise the contract. He buys the asset for an amount of K and he could sell that same asset immediately on the market for an amount of S_T. For this case, the payoff is strictly positive and given by $S_T - K$. Shortly, for the payoff function we can write

$$\left.\begin{array}{l}\text{if } S_T < K \Longrightarrow \text{ don't exercise, payoff is equal to } 0 \\ \text{else } S_T \geq K \Longrightarrow \text{ exercise, payoff is equal to } S_T - K\end{array}\right\} = \max\{S_T - K, 0\} = (S_T - K)^+ .$$

In Figure 3, the payoff function is sketched for an arbitrary $K > 0$.

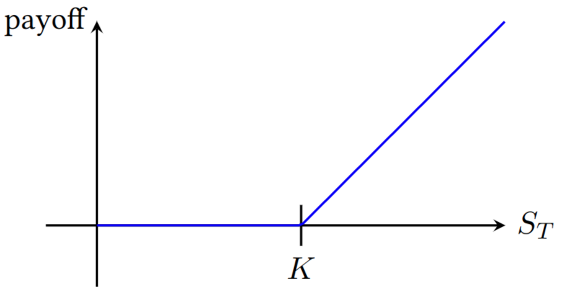


Figure 3: The pay-off diagram of a European call option: $(S_T - K)^{+}$[1].

For a put option, a similar way of thinking applies. For this type of option, it is not wise to exercise the contract if the asset value is greater than or equal to the strike price, i.e., $S_T \geq K$. If someone exercises, this person would sell the asset for K, while it is worth more on the market. Therefore, the put option is not exercised and the payoff is zero. On the other hand, if $S_T < K$, the contract holder should exercise the contract. In this case, he sells the asset for an amount of K and he could buy that same asset immediately on the market for an amount of S_T. Therefore, the payoff is strictly positive and given by $K - S_T$. In short, for this payoff function, we can write

$$\left.\begin{array}{l}\text{if } S_T < K \Longrightarrow \text{ exercise, payoff is equal to } K - S_T \\ \text{else } S_T \geq K \Longrightarrow \text{ don't exercise, payoff is equal to } 0\end{array}\right\} = \max\{K - S_T, 0\} = (K - S_T)^+ \,.$$

In Figure 4, the payoff function is sketched for an arbitrary $K > 0$.

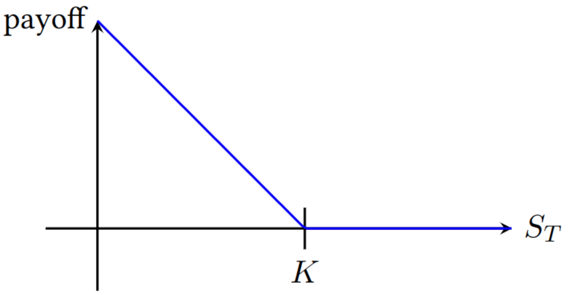


Figure 4: The pay-off diagram of a European put option: $(K - S_T)^{+}$[2].

1.1.4 Put-call parity

A beautiful equality that allows us to express the value of a call option in terms of a put option, the value of the stock, and the (discounted) strike price is given by the put-call parity. One condition that needs to be satisfied to have equality is that the underlying asset, the strike price K, and the time to maturity $T - t$ are identical for the two options.

[1] Image taken from [3].

[2] Image taken from [3].

Theorem 1 (Put-call parity). *Let S be some asset, $V_i(S,t)$ the value of an option at time t for $i \in \{call,\ put\}$ with S being the underlying asset, K the strike price and $T-t$ the time to maturity. Moreover, let r be the interest rate and assume that r is constant. If $S(t)$ is the price of the asset at time t, then the put-call parity is given by:*

$$V_{call}(t,S) + Ke^{-r(T-t)} = V_{put}(t,S) + S(t).$$

The proof is omitted. The interested reader who wants to see the proof is referred to page 55 of [2].

1.2 The Black-Scholes equation

Bonds, call, and put options all have in common that they are financial contracts between two parties. An interesting task is to find a price for these financial contracts that is considered fair for both parties. The Black-Scholes PDE is one of the most famous PDEs used for pricing European call and put options.

1.2.1 Background Information

In 1973, the Black-Scholes partial differential equation made its first appearance. It was developed by three mathematicians: Fischer Black, Myron Scholes, and Robert C. Merton. Black and Scholes published an article containing the formula in the Journal of Political Economy [4]. The writers of [5] claim that the model showed that mathematics plays a significant role in finance. Since the article of Black and Scholes was very focused on the economic part of the model, Merton wrote a paper in the same year, in which he paid attention to the mathematical understanding of the model [6]. In 1997, two after Black passed away, Scholes and Merton were awarded the Nobel Memorial Prize in Economic Sciences for their work on this model.

Figure 5: From left to right: Robert Merton, Myron Scholes, and Fischer Black.[3]

[3] Image taken from `https://gfmag.com/features/5-black-scholes-merton-and-algorithms/`.

As stated in [7], the Black-Scholes formula is a groundbreaking equation. It is the most effective way of pricing options. Using the Black-Scholes formula, one can very quickly compute a so-called 'fair' price for both the option buyer and seller. John Lister states another advantage of the BS model on the website of Smart Capital Mind [8]: *"The main advantage of the model is that it works entirely on objective figures rather than human judgment."*

The popularity of this model increased rapidly in the financial industry during the seventies and, although there are more sophisticated models at hand today, the Black-Scholes model still remains important for option valuation.

1.2.2 Preliminaries

Some mathematical tools must be introduced before presenting the Black-Scholes PDE. The most important concepts are that of a Brownian motion (also called a Wiener process) and a stochastic differential equation (SDE). To understand these concepts, we start with the definition of a stochastic process. In this definition, we mention some measure theoretical notions, but we will not delve too much into these notions. The definition below is cited from [9].

Definition 5 (**Stochastic process**). *Let I be an ordered set (in our setting this is often $\mathbb{R}_{\geq 0}$), $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space, and $(E, \mathscr{G})$ a measurable space. A stochastic process is a collected of random variables $X = \{X_i; i \in I\}$ such that for each fixed $i \in I$, X_i is a random variable from $(\Omega, \mathcal{F}, \mathbb{P})$ to $(E, \mathscr{G})$. The set Ω is the sample space, where E is the state space of the stochastic process X_i.*

The following definition is that of a Brownian motion, cited from [10]. As described in [11], the term 'Brownian motion' is named after the Scottish botanist Robert Brown. Almost a century later, in 1918, the American Jew Norbert Wiener, both a mathematician and philosopher, succeeded in writing down a rigorous mathematical formulation. For this reason, a Brownian motion is also called a Wiener process. [12] provides an interesting view of the history of Brownian motion.

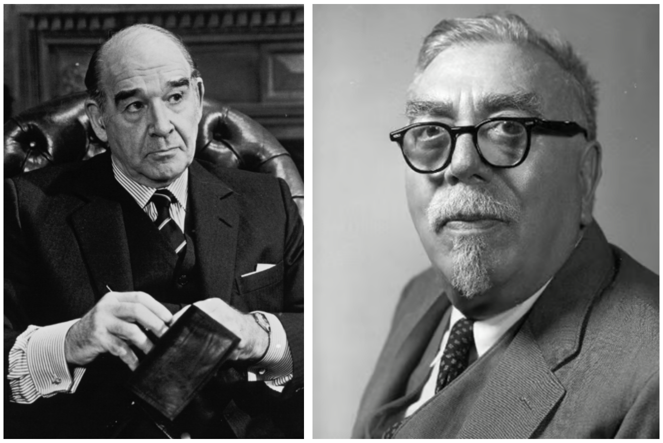


Figure 6: Robert Brown (l) and Norbert Wiener[4](r).

Definition 6 (**Brownian motion**). *A real-valued stochastic process* $\{W_t : t \geq 0\}$ *is called a (standard) Brownian motion if the following holds:*

- $W_0 = 0$,
- *the process has independent increments, i.e., for all times* $0 \leq t_1 \leq t_2 \leq \ldots \leq t_n$ *the increments* $W_{t_n} - W_{t_{n-1}}, W_{t_{n-1}} - W_{t_{n-2}}, \ldots, W_{t_2} - W_{t_1}$ *are independent random variables,*
- *for all* $t \geq 0$ *and* $h > 0$*, the increment* $W_{t+h} - W_t$ *is normally distributed with expectation zero and variance* h*. This is denoted by* $\mathcal{N}(0, h)$*,*
- *almost surely, the function* $t \mapsto W_t$ *is continuous.*

Brownian motion can be applied in all kinds of fields. Not only is it used for option pricing, but it also has applications in biology [13], chemistry [14], and physics [15] (obviously, these three fields have a little overlap). It is one of the main ingredients used in the definition of a stochastic differential equation, which is derived from [2].

Definition 7 (**Stochastic Differential Equation**). *Let* $\{X_t, t \geq t_0\}$ *be a stochastic process. A stochastic differential equation (SDE) is a differential equation that incorporates a stochastic process. An SDE can be represented as*

$$dX_t = \bar{\mu}(X_t, t)\, dt + \bar{\sigma}(X_t, t)\, dW_t \text{ with } X_{t_0} = X_0.$$

Each of the terms used above has the following meaning:

- X_0 *is the starting value,*
- dX_t *denoted the change in the stochastic process* X_t*,*
- $\bar{\mu}(X_t, t)$ *is the deterministic drift term,*
- $\bar{\sigma}(X_t, t)$ *is the stochastic volatility term (also called the diffusion term),*
- W_t *is a (standard) Brownian motion.*

To find a stochastic differential equation that arises from a certain (green) derivative, an indispensable tool that is needed is Itô's lemma, named after the Japanese mathematician Kiyosi Itô, who contributed a lot to the study of SDEs around the 1940s. The lemma can be found below and is cited from [2].

[4] Images taken from `https://www.imdb.com/name/nm0114533/` and `https://www.onthisday.com/people/norbert-wiener`.

Figure 7: Kiyoshi Itô[5]

Lemma 1 (Itô's lemma). *Suppose a stochastic process X_t follows the Itô dynamics, given by*

$$dX_t = \bar{\mu}\left(t, X_t\right) dt + \bar{\sigma}\left(t, X_t\right) dW_t \text{ with } X_{t_0} = X_0,$$

where drift $\bar{\mu}\left(t, X_t\right)$ and diffusion $\bar{\sigma}\left(t, X_t\right)$ satisfy the standard Lipschitz conditions on the growth of these functions. Let $g(t, X)$ be a function of $X = X(t)$ and time t, with continuous partial derivatives

$$\frac{\partial g}{\partial X}, \frac{\partial^2 g}{\partial X^2}, \text{ and } \frac{\partial g}{\partial t}.$$

A stochastic variable $Y(t) := g(t, X)$ has the following dynamics, governed by the same Brownian motion W_t, i.e.,

$$dY_t = \left(\frac{\partial g}{\partial t} + \bar{\mu}\left(t, X\right) \frac{\partial g}{\partial X} + \frac{1}{2} \frac{\partial^2 g}{\partial X^2} \bar{\sigma}^2\left(t, X\right)\right) dt + \frac{\partial g}{\partial X} \bar{\sigma}\left(t, X\right) dW_t.$$

A detailed version of the proof is provided in [16].

1.2.3 The PDE and its boundary conditions

The Black-Scholes equation is given by

$$\begin{cases} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \\ \forall S \geq 0 : V(T, S) = F(S). \end{cases}$$

[5] Image taken from `https://nl.wikipedia.org/wiki/Kiyoshi_Ito`.

Using this notation, we have that

- $S = S(t)$ is the stock price at time $t \in [0, T]$,
- $V = V(S, t)$ is the value of the option on a stock with value S at time t.
- σ is the volatility of the stock,
- r is the interest rate.

This PDE also comes with a boundary condition. Often, it is the case that the initial condition (the function value at $t = 0$) is given. For the Black-Scholes PDE, on the other hand, we are given the terminal condition (i.e., the function value at $t = T$). We have already encountered these boundary conditions, namely, if F is the payoff function, we can distinguish the cases where we have a call and put option.

- For a call option, we have that $F(S) = (S_T - K)^+ := \max\{S_T - K, 0\}$.
- For a put option, we have that $F(S) = (K - S_T)^+ := \max\{K - S_T, 0\}$.

The boundary conditions above are so-called **Dirichlet** boundary conditions, i.e., the function value at the boundary is known. There are also cases where not the function value itself, but the derivative of the function at the boundary is known. These are called **Neumann** boundary conditions. Sometimes, the two types of boundary conditions are used together. The interested reader is invited to take a look at [17].

The Black-Scholes PDE can be derived in multiple ways, see for example [18] or [19].

1.2.4 Analytical solution

The Black-Scholes equation can be solved analytically. It goes beyond the scope of this thesis, but those who are interested are invited to see the proof of the next Theorem in [3].

Theorem 2. *Write*

$$d_1 = \frac{\log\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

and

$$d_2 = \frac{\log\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}.$$

Moreover, let Φ be the cumulative distribution function of the standard normal distribution. Then, the solution of the Black-Scholes PDE for a call option is given by

$$V_{call}(t, S) = S(t)\Phi\left(d_1\right) - Ke^{-r(T-t)}\Phi\left(d_2\right).$$

Using the put-call parity, a similar expression can be found for V_{put}:

$$V_{put}(t, S) = Ke^{-r(T-t)}\Phi\left(-d_2\right) - S(t)\Phi\left(-d_1\right).$$

1.2.5 Assumptions and disadvantages of Black-Scholes

In the Black-Scholes model, we assume that no **arbitrage** opportunities are available. Simply put, no arbitrage means nobody can make a profit without taking a risk. A formal definition of arbitrage is stated below and is derived from [3].

Definition 8 (**Arbitrage**). *An investment strategy with value process V is called an arbitrage if the following all hold:*

- $V_0 \leq 0$ *(zero initial cost),*
- $\mathbb{P}\left(V_1 \geq 0\right) = 1$ *(no losses with certainty),*
- $\mathbb{P}\left(V_1 > 0\right) > 0$ *(gain with positive probability).*

Moreover, we assume that the asset price S follows a log-normal distribution, i.e. $g(t, S_t) := \ln(S_t)$ follows a normal distribution. Furthermore, the asset price satisfies a stochastic differential equation, see Definition 7. The formula for the dynamics of the asset price S is given by

$$dS_t = rS_t dt + \sigma S_t dW_t.$$

Some advantages of the Black-Scholes model were already stated in a previous section. Although the Black-Scholes formula forms the building blocks of option pricing theory, a few drawbacks are worth mentioning.

- The Black-Scholes model does not take transaction costs into account despite being a common phenomenon in practice. The same can be said about dividend payments: they are considered vacant.
- In the real world, trading is a discrete process, but in the Black-Scholes model, we assume this is a continuous process.
- The values of the interest rate r and the volatility σ are assumed to be constant, but the values of these parameters change all the time.

1.2.6 Link with the heat equation

As already stated before, the Black-Scholes equation is given by

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

for $V(S, t)$. It turns out that this equation is equivalent to the well-known heat equation, which is given by

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2},$$

for $y(x, \tau)$. The details of this statement are provided in Appendix A.

1.3 Sustainable Finance

The following quote comes from Jeffrey D. Sachs [20], an American economist, professor, and director of the Earth Institute at Columbia University: *"Since the Industrial Revolution, finance has been a powerful enabler of human progress. The purpose of the global financial system is to allocate the world's savings to their most productive uses. When the system works properly, these savings are channeled into investments that raise living standards; when it malfunctions, as in recent years, savings are channeled into real-estate bubbles and environmentally harmful projects, including those that exacerbate human-induced climate change."*

The researchers of [20] claim that financial institutions mostly show more interest in fossil fuel projects than green projects, mainly because there are still several risks associated with these new technologies and they offer a lower rate of return. This statement is from 2017 and in the meantime, terms like 'green finance' and 'sustainable development' have become increasingly more important. Luckily, they have also become increasingly more popular [21]. This holds both for companies and governments. Copying behavior is an important factor that can result in an acceleration of taking measures against pollution. For example, governments putting a price on greenhouse gas emissions is an increasingly occurring phenomenon. In [22] the argument is made that putting a price on carbon provides incentives to opt for less carbon-intensive alternatives while still leaving it to markets to determine the best and most cost-effective technologies and solutions. In the same paper, a study revealed that governments are more likely to opt for carbon pricing policies when their trading partners or competitor countries that export goods to the same market also do so. This means many governments tend to jump on the bandwagon after only some governments implement a green policy.

1.4 Green bonds

A green bond is an example of a green financial derivative. It is a relatively new type of bond [23] and is considered by the writer of [24] as one of the most prominent innovations in the field of sustainable finance over the past decade. The green bond market was invented not earlier than in 2007 [25] but is growing rapidly [21]. In the literature, there is some inconsistency in the definition of a green bond. [23] describes a green bond as *"any type of bond instrument where the proceeds will be exclusively applied to finance or re-finance, in part or in full, new or/and existing eligible green projects."* In [21] it is stated that, according to Bloomberg, *"green bonds are financial securities to finance projects to minimize the impact of greenhouse gas emissions"*, whereas [26] informs us that *"green bonds are earmarked to finance only projects with environmental benefits"*. Unfortunately, no clear definition of an 'environmentally beneficial' project is available. This can be problematic, and [26] argues that this leads to market participants using their own (different) standards. One of the disadvantages of the haziness is that the 'greenness' of a bond can be exaggerated. This is called **greenwashing**.

1.4.1 Classes of green bonds

In [26], the researchers dive into a framework for classifying green bonds designed by the International Capital Market Association (ICMA). This framework considers four criteria to classify a bond as 'green'. The criteria are given below.

- A bond is green if the Use of Proceeds is satisfied. According to ICMA, this is if the bond issuer earmarks the bond proceeds to finance eligible green projects described in their legal documentation.
- The Process for Project Evaluation and Selection principle is satisfied if the issuers communicate
  - what the objectives of the green bond project are,
  - what makes the project eligible, and
  - what the associated environmental and social risks are.
- The Management of Proceeds principle is satisfied if the bond proceeds are managed and tracked within the company's financial structure.
- The Reporting Principle is fulfilled if a company reports on their green bond Use of Proceeds and the projects to which funds have been allocated in the final report.

We can divide all green bonds into three groups using these indicators. The first group is formed by the green bonds that only satisfy the Use of Proceeds principle. This is considered the lowest form of greenness. The green bonds that fulfill all four ICMA principles form the second highest form of greenness. If, in addition, a green bond also has received the review of a third party (an effective measure to prevent greenwashing), it is granted the rank of highest greenness.

The above classification was used up to the end of 2023. In November 2023, the European Green Bond Standard (EUGBS) was introduced as part of the European Green Deal investment plan [27]. The subject matter is as follows. This regulation

- lays down uniform requirements for issuers of bonds who wish to use the designation 'European Green Bond' or 'EuGB' for their bonds that are made available to investors in the Union,
- establishes a system to register and supervise external reviewers of European Green Bonds, and
- provides optional disclosure templates for bonds marketed as environmentally sustainable and for sustainability-linked bonds in the Union.

1.4.2 Comparing green bonds to ordinary bonds

Although not everyone has worked with the same definition of a green bond, they share a willingness to move to a 'greener' world. Making the green bond market successful is important, since [21] claims that financing via green bonds is an essential mechanism to support sustainable development. To make this market a success, both issuers and investors of green bonds should be satisfied with the returns and safety of such security. For an investor, there is a direct financial incentive to invest in a green bond if this bond provides better returns compared to other bonds [24]. It turns out that there are differences between green bonds and ordinary bonds (also called brown bonds) regarding yields. Sadly, in [28], it is shown that the returns on brown bonds are on average higher than for green bonds. The writers claim that *"the market penalizes green bonds to a higher degree than brown bonds."* Similar results follow from [24], where the researchers discovered that the yields of green bonds are on average two basis points (abbreviated as bps, 1 bps correspond to 0.01%)

lower than those of comparable brown bonds.

In search of an explanation for this phenomenon, it is worthwhile to mention that it is not only beneficial for issuers of bonds but also for investors to have a green bond instead of a brown bond. Some investors are even willing to pay a premium to hold a green bond instead of a brown bond. This premium is called a **greenium**. According to the researchers, the most reasonable explanation for this phenomenon is the high demand for and limited supply of green bonds. A company wants to be involved in sustainable finance since it shows the organization's efforts to

- secure legitimacy in the face of societal level pressures to demonstrate sustainable business practices,
- demonstrate accountability to identifiable stakeholders with sustainability demands on the organization,
- conform to sustainability practices of other like-organizations facing the same institutional level pressures to engage with sustainability.

The writers of [21] recommend future studies to focus on new developments in the green bond market and check different market sentiments' effects of the premium of green bonds, especially since this market is in the evolutionary phase.

1.5 Pricing a green bond

In practice, many different types of green bonds exist on the market. For this master thesis, we will consider a green bond model that considers both the interest rate and the carbon price. This model is derived in [29] and can be found in Chapter 4.1. It is already stated that the price needs to be fair for the buyer and the seller of a bond; this is a delicate task. In the next chapters, we will introduce mathematical theory that can be used to determine the price of a green bond. This involves numerical mathematics. It should be mentioned that determining a fair price is not the main aim of this thesis, we will evaluate stability, which plays a crucial role in numerical mathematics.

2 Numerical Analysis

When we are solving an equation, the first thing that we want to do is try to find an analytical solution. Such a solution is also referred to as a 'closed-form solution'. Unfortunately, when solving PDEs and SDEs, this is hardly ever possible. In [30], it is explained why most of the PDEs can not be solved by hand. Due to this, we have to settle for a numerical solution instead of an analytical solution. Numerical solutions can be obtained by discretizing the domain of interest and trying to find the value of the function on the nodes in the grid. This solution is often not the exact solution, but an approximated solution. Using the finite difference method, we obtain a system of linear equations, denoted by $A\mathbf{u} = \mathbf{f}$. There are multiple ways to solve such a system. A distinction can be made between direct solution methods and iterative solution methods. For this thesis, we focus on both methods.

If matrix A were to be invertible, we could find the solution $\mathbf{u}$ by just multiplying the right-hand side vector $\mathbf{f}$ with A^{-1}, i.e., $\mathbf{u} = A^{-1}\mathbf{f}$. If computing A^{-1} was easy, iterative solution methods would not have been necessary. The problem is that the computation of an inverse of a matrix is 'expensive'. That is, computing the inverse of a $n \times n$ matrix is, from a computational viewpoint, a heavy task. Take for example Gauss-Jordan elimination: computating A^{-1} using this method costs $O\left(n^3\right)$ operations.

2.1 Preliminaries

This section contains all the definitions and theorems that might be useful for the research part of this thesis.

2.1.1 Some definition about matrices

When analyzing iterative solution methods for systems of linear equations, we come across matrices. Good knowledge of matrices is necessary. In this section, some key results about matrices are presented. The definitions and theorem are all cited from [31].

Definition 9 (**Matrix norm**). *Given $1 \leq p < \infty$, the p-norm of a matrix $R \in \mathbb{R}^{m \times n}$, denoted as $\|R\|_p$, is defined by*

$$\|R\|_p = \sup_{\mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\|R\mathbf{u}\|_p}{\|\mathbf{u}\|_p}.$$

In the case of $p = 1$, $p = 2$ and $p = \infty$, the following expressions exist that allow us to compute the matrix p-norm in practice

$$\begin{aligned} \|R\|_1 &= \max_{1 \leq j \leq n} \sum_{i=1}^{m} |r_{ij}| \quad \text{maximum absolute column sum}, \\ \|R\|_2 &= \sqrt{\max_{1 \leq i \leq n} \lambda_i \left(R^\top R\right)} = \sqrt{\lambda_{\max} \left(R^\top R\right)}, \\ \|R\|_\infty &= \max_{1 \leq i \leq m} \sum_{j=1}^{n} |r_{ij}| \quad \text{maximum absolute row sum}. \end{aligned}$$

From linear algebra, we know that all vector norms are equivalent. The same result holds for matrix norms. That is, if $\|\cdot\|_p$ and $\|\cdot\|_q$ are two matrix norms on $n \times n$ matrices, then there exists constant α and β such that

$$\forall A \in \mathbb{R}^{n\times n} : \alpha\|A\|_p \leq \|A\|_q \leq \beta\|A\|_p.$$

We move on with the definition of the spectral radius. The (speed of) convergence of iterative solution methods depends on the value of the spectral radius.

Definition 10 (**Spectral radius**). *Let $A \in \mathbb{R}^{n\times n}$ and let $\sigma(A)$ be the set of eigenvalues of A. The spectral radius $\rho(A)$ of matrix A is defined as*

$$\rho(A) = \max_{i=1,\dots,n} \left\{|\lambda_i| : \lambda_i \in \sigma(A)\right\}.$$

It is not always the case that $\rho(A) \in \sigma(A)$. Think for example about the case where we have negative or complex eigenvalues. In general, it is difficult to compute the spectral radius, but it is possible to find an upper bound that is much easier to compute. This upper bound is given in the next theorem. It can be seen as a link between the spectral radius and a matrix norm.

Theorem 3. *Let $\|\cdot\|$ be any multiplicative norm. That is, for all $1 \leq p < \infty$ and for all $A \in \mathbb{R}^{m\times q}$ and $B \in \mathbb{R}^{q\times n}$ we have that*

$$\|AB\|_p \leq \|A\|_p\|B\|_p.$$

Then the following inequality holds:

$$\rho(A) \leq \|A\|.$$

Proof. Assume $(\lambda, \mathbf{u})$ to be any eigenvalue-eigenvector pair of A. Then $A\mathbf{u} = \lambda\mathbf{u}$, and by making use of the sub-multiplicative property we find

$$|\lambda|\|\mathbf{u}\| = \|\lambda\mathbf{u}\| = \|A\mathbf{u}\| \leq \|A\|\|\mathbf{u}\| \implies |\lambda| \leq \|A\|.$$

Since the eigenvalue λ was arbitrary, the proof is finished. □

One of the iterative solution methods we will analyze is the Conjugate Gradient method (see Chapter 2.7.1). The next theorem forms a link between the convergence of this method and the condition number of a matrix. The definition of the condition number can be found below.

Definition 11 (**Condition number**). *The condition number measured in p-norm $\kappa_p(A)$ of an invertible $n \times n$ matrix A is defined as*

$$\kappa_p(A) = \|A\|_p\|A^{-1}\|_p.$$

Theorem 4. *The condition number measured in p-norm for any invertible $n\times n$ matrix A is greater than or equal to 1, i.e.,*

$$\kappa_p(A) \geq 1.$$

Proof. Let A be a matrix that satisfies the conditions of Theorem 4 and let p be any multiplicative norm. Using the multiplicative property, we find that

$$\begin{aligned}\kappa_p(A) &= \|A\|_p \|A^{-1}\|_p \\ &\geq \|AA^{-1}\|_p \\ &= \|I_n\|_p \\ &= 1.\end{aligned}$$

□

For any $n \times n$ matrix A, the spectral radius and the matrix norm depend on the set of eigenvalues $\sigma(A)$. Computing all the eigenvalues is often too difficult, but the next theorem allows us to find an approximation (i.e., lower or upper bound) of the eigenvalues.

Theorem 5 (**Gershgorin's Circle Theorem**). *Let $A \in \mathbb{R}^{n\times n}$. If $\lambda \in \sigma(A)$, then λ is located in one of the n closed disks in the complex plane that has centre a_{ii} and radius*

$$\rho_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|,$$

i.e.,

$$\lambda = \sigma(A) \Longrightarrow \exists i \textit{ such that } |a_{ii} - \lambda| \leq \rho_i.$$

It is already mentioned that for this master's thesis, the systems of linear equations are of the form $A\mathbf{u} = \mathbf{f}$. Sometimes matrix A is symmetric and positive definite, abbreviated as SPD. The next definition dives into this property.

Definition 12 (**SPD matrix**). *A matrix $A \in \mathbb{R}^{n\times n}$ is called symmetric positive definite (SPD) if and only if A is symmetric (that is, $A^\top = A$) and*

$$\forall \mathbf{u} \in \mathbb{R}^n \backslash \{\mathbf{0}\} : \mathbf{u}^\top A \mathbf{u} > 0.$$

If we allow equality in the above inequality, we call A symmetric positive semi-definite.

If a $n \times n$ matrix A is SPD, we have that all eigenvalues are real and positive. In particular, 0 is not an eigenvalue; thus, A is invertible. We can define $\|\cdot\|_A$, given by

$$\forall \mathbf{u} \in \mathbb{R}^n : \|\mathbf{u}\|_A := \mathbf{u}^\top A \mathbf{u}.$$

It turns out that $\|\cdot\|_A$ satisfies all the properties of being a norm. Therefore, $\|\cdot\|_A$ is a norm, which is called the norm induced by matrix A.

In Section 2.7, we will consider several iterative solution methods. All these methods are so-called Krylov subspace methods. Therefore, we present the definition of a Krylov subspace.

Definition 13 (**Krylov subspace**). *The Krylov subspace of dimension k corresponding to a matrix A and a vector $\mathbf{v}$ is given by*

$$K^k := \textit{span}\left\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\right\}.$$

2.2 Finite Difference Discretization

At the beginning of this chapter, it was already stated that, unlike the Black-Scholes PDE, most PDEs and SDEs cannot be solved analytically. Solving such equations on a given domain is still possible, just not analytically. Therefore, we move from analytical solutions to numerical solutions. To obtain such a solution, we first discretize the (multidimensional) domain. Later in this chapter, we will use the heat equation as an example. This PDE is given by

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},$$

for $0 \leq x \leq S_{\max}$ and $0 \leq t \leq T$, subject to

- the initial condition $u(x, 0) = g(x)$, and
- the two boundary conditions $u(0, t) = a(t)$ and $u\left(S_{\max}, t\right) = b(t)$.

We will discretize the domain both in the space direction x (or S, the initial value of the option) and the time direction t. The domain of the space is then given by $[0, S_{\max}] \times [0, T]$.

In the next sections, we will elaborate on how we can discretize the domain. In Section 2.2.1, we will dive into the discretization of space and Section 2.3.1 is about the discretization of the time direction.

2.2.1 Discretization of space

For a continuous and smooth function $y : \mathbb{R} \to \mathbb{R}$, the first derivative with respect to x is a well-known result described by the following limit.

$$\lim_{h \to 0} \frac{y(x+h) - y(x)}{h} = \frac{dy}{dx}(x) = y'(x).$$

Similarly, for the second derivative of y with respect to x, we find another limit given by:

$$\lim_{h \to 0} \frac{y(x-h) - 2y(x) + y(x+h)}{h^2} = \frac{d^2 y}{dx^2}(x) = y''(x).$$

In the world of numerical mathematics, we are not in the position to take the limit of h going to zero. We can decide to choose h to be 'small', which means that the above equalities become approximations:

$$\begin{aligned} \frac{y(x+h) - y(x)}{h} &\approx y'(x) \iff y(x+h) - y(x) \approx h y'(x), \\ \frac{y(x-h) - 2y(x) + y(x+h)}{h^2} &\approx y''(x) \iff y(x-h) - 2y(x) + y(x+h) \approx h^2 y''(x). \end{aligned}$$

This value of h can be considered as the distance between two grid points in, for example, the space direction. For this master thesis, we assume that this distance is the same for any two grid points, i.e., all the points are equidistant. If this holds, we speak of a uniform grid. Note that this doesn't have to be the case. The interested reader is invited to consult [32], where a non-uniform grid is used to find a numerical solution to the Black-Scholes equation without boundary conditions.

Suppose we have $M+1$ grid points in the spatial direction, which runs from 0 to $S_{\max}$, where the space between every two consecutive points is h, i.e.,

$$X = \left\{ x_m \mid x_m = (m-1)h;\ h = \frac{1}{M},\ 1 \leq m \leq M+1 \right\},$$

then we can simply write $y(x_m) = y_m$. Below, we introduce several finite difference operators. We will derive the Taylor series for each operator since this provides useful information for local accuracy.

Forward difference: $\Delta y_m = y_{m+1} - y_m$. Using the Taylor series and adding the left-hand and right-hand sides, we find

$$\begin{array}{rl} y_{m+1} &= y_m + hy'_m + \frac{1}{2}h^2 y''_m + \frac{1}{6}h^3 y'''_m + \dots \\ -y_m &= -y_m \\ \hline y_{m+1} - y_m &= hy'_m + \frac{1}{2}h^2 y''_m + \frac{1}{6}h^3 y'''_m + \dots. \end{array} \tag{1}$$

Backward difference: $\nabla y_m = y_m - y_{m-1}$. Using the Taylor series and adding the left-hand and right-hand sides, we find

$$\begin{array}{rl} y_m &= y_m \\ -y_{m-1} &= -y_m - (-h)y'_m - \frac{1}{2}(-h)^2 y''_m - \frac{1}{6}(-h)^3 y'''_m + \dots \\ \hline y_{m+1} - y_m &= hy'_m - \frac{1}{2}h^2 y''_m + \frac{1}{6}h^3 y'''_m + \dots. \end{array} \tag{2}$$

Half central difference: $\delta y_m = y_{m+\frac{1}{2}} - y_{m-\frac{1}{2}}$. Using the Taylor series and adding the left-hand and right-hand sides, we find

$$\begin{array}{rl} y_{m+\frac{1}{2}} &= y_m + \left(\frac{1}{2}h\right) y'_m + \frac{1}{2}\left(\frac{1}{2}h\right)^2 y''_m + \frac{1}{6}\left(\frac{1}{2}h\right)^3 y'''_m + \dots \\ -y_{m-\frac{1}{2}} &= -y_m - \left(-\frac{1}{2}h\right) y'_m - \frac{1}{2}\left(-\frac{1}{2}h\right)^2 y''_m - \frac{1}{6}\left(-\frac{1}{2}h\right)^3 y'''_m + \dots \\ \hline y_{m+\frac{1}{2}} - y_{m-\frac{1}{2}} &= hy'_m + \frac{1}{24}h^3 y'''_m + \dots. \end{array} \tag{3}$$

Second order central difference: $\delta^2 y_m = y_{m+1} - 2y_m + y_{m-1}$. Using the Taylor series and adding the left-hand and right-hand sides, we find

$$\begin{aligned}
y_{m+1} &= y_m + hy'_m + \frac{1}{2}h^2 y''_m + \frac{1}{6}h^3 y'''_m + \frac{1}{24}h^4 y''''_m + \dots \\
-2y_m &= -2y_m \\
y_{m-1} &= y_m + (-h)y'_m + \frac{1}{2}(-h)^2 y''_m + \frac{1}{6}(-h)^3 y'''_m + \frac{1}{24}(-h)^4 y''''_m + \dots
\end{aligned}$$

$$y_{m-1} - 2y_m + y_{m+1} = h^2 y''_m + \frac{1}{12}h^4 y''''_m + \dots. \tag{4}$$

Average: $\mu = \frac{1}{2}(y_{m+\frac{1}{2}} + y_{m-\frac{1}{2}})$. Using the Taylor series and adding the left-hand and right-hand sides, we find

$$\begin{aligned}
\frac{1}{2}y_{m+\frac{1}{2}} &= \frac{1}{2}y_m + \frac{1}{2}\left(\frac{1}{2}h\right) y'_m + \frac{1}{2}\left(\frac{1}{2}h\right)^2 y''_m + \dots \\
\frac{1}{2}y_{m-\frac{1}{2}} &= \frac{1}{2}y_m + \frac{1}{2}\left(-\frac{1}{2}h\right) y'_m + \frac{1}{2}\left(-\frac{1}{2}h\right)^2 y''_m + \dots
\end{aligned}$$

$$\frac{1}{2}\left(y_{m+\frac{1}{2}} + y_{m-\frac{1}{2}}\right) = y_m + \frac{1}{4}h^2 y''_m + \dots. \tag{5}$$

2.3 Time-stepping methods

This section is cited partly from [33].

2.3.1 Discretization of time

Suppose we have $N+1$ grid points in the time direction, which runs from 0 to T, where the space between every two consecutive points is Δt, i.e.,

$$Y = \left\{ t_n \mid t_n = (n-1)\Delta t;\ \Delta t = \frac{1}{N},\ 1 \le n \le N+1 \right\}.$$

We will give three different elementary time-stepping methods. Each method has advantages and disadvantages that will be described along the way. To make life simple, we introduce the time integration methods for the following scalar first-order differential equation

$$\begin{cases} \frac{dy}{dt} = y' = f(t,y), t > t_0, \\ y\left(t_0\right) = y_0. \end{cases}$$

The solution of the above differential equation can be found by integrating the function f with respect to time t:

$$y(t) = y(t_0) + \int_{t_0}^{t} f\left(\tau, y(\tau)\right) d\tau.$$

Using our discretized time interval $[0, T]$, we can compute the solution of the differential equation step by step:

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f\left(t, y(t)\right) dt.$$

The next subsections will present several time-stepping methods, including background information.

2.3.2 Forward Euler method

The Forward Euler method is one of the most ordinary time-stepping methods. It is named after the Swiss mathematician Leonhard Euler, who wrote about this method in his book Institutionum calculi integralis [34], published in the second half of the 18th century.

Figure 8: Leonhard Euler[6]

This method is also called the left Rectangle rule, and this is for a good reason. To make a step from $t = t_n$ to $t = t_{n+1}$, we compute the value of f for given t_n and y_n (which are both known at time t_n). From here, we pretend that the function value remains constant on the interval $[t_n, t_{n+1}]$. In Figure 9, we see that by performing this method, we obtain small rectangles where the height is determined by the function value in the left endpoint in the interval $[t_n, t_{n+1}]$.

[6] Image taken from `https://nl.wikipedia.org/wiki/Leonhard_Euler`.

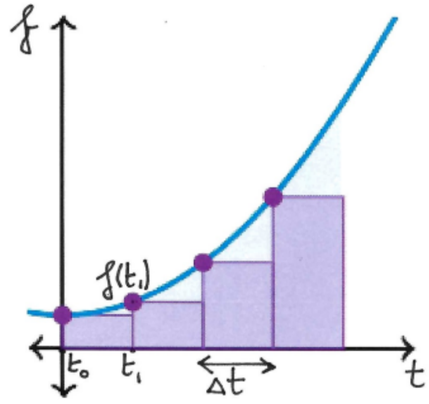


Figure 9: Graphical representation of the Forward Euler method[7]. Time is on the horizontal axis, and the function value f is on the vertical axis.

The approximation for y_{n+1} becomes as follows:

$$y_{n+1} \approx y_n + (t_{n+1} - t_n) f(t_n, y_n) = y_n + \Delta t f(t_n, y_n).$$

The numerical approximation at time t_{n+1} is denoted by w_{n+1} and is given by

$$w_{n+1} = w_n + \Delta t f(t_n, w_n).$$

Note that at time t_n, the value of w_n and, thus, the value of $f(t_n, w_n)$ are known explicitly. For this reason, the Forward Euler is called an explicit method.

2.3.3 Backward Euler Method

The Backward Euler method can also be represented by using rectangles, but for this method, the height of a rectangle is determined by the function value of f for $y = y_{n+1}$ and $t = t_{n+1}$. Therefore, this method is also called the right Rectangle rule. After determining $f(t_{n+1}, y_{n+1})$, again we pretend that the function value remains constant on the interval $[t_n, t_{n+1}]$. This is illustrated in Figure 10.

[7] Image taken from `https://www.khanacademy.org/math/ap-calculus-ab/ab-integration-new/ab-6-2/a/left-and-right-riemann-sums`.

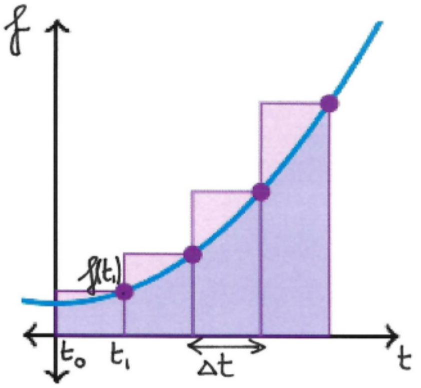


Figure 10: Graphical representation of the Backward Euler method[8]. Again, time is on the horizontal axis, and the function value f is on the the vertical axis.

The numerical approximation at time t_{n+1} is given by

$$w_{n+1} = w_n + \Delta t f\left(t_{n+1}, w_{n+1}\right).$$

Note that at time t_n, the value of w_{n+1} and, thus, the value of $f\left(t_n, w_{n+1}\right)$ are not known explicitly. This means we are left with an equation that should be solved for w_{n+1}. Solving such an equation can be done using Newton's method or a different fixed point method. If such an equation occurs, we call a time-stepping method implicit.

2.3.4 Crank-Nicolson

The Crank-Nicolson method (also called the Trapezoidal Rule) was developed by two British mathematicians John Crank and Phyllis Nicolson. They published a paper on this topic in 1946 [35].

Figure 11: John Crank and Phyllis Nicolson[9].

[8] Image taken from https://www.khanacademy.org/math/ap-calculus-ab/ab-integration-new/ab-6-2/a/left-and-right-riemann-sums.

The Crank-Nicolson method (also called the Trapezoidal method) can be interpreted as a mix between the Forward and Backward Euler method. We do not assume the function f to be constant on the interval $[t_n, t_{n+1}]$. Instead, the function f is believed to grow or decay linearly from $f(t_n, y_n)$ to $f(t_{n-1}, y_{n-1})$. This is illustrated in Figure 12 below. It should be mentioned that in this figure, the rectangles do not have the same base whereas we already stated that the time steps are all of size Δt. Nevertheless, it shows very well how the method works.

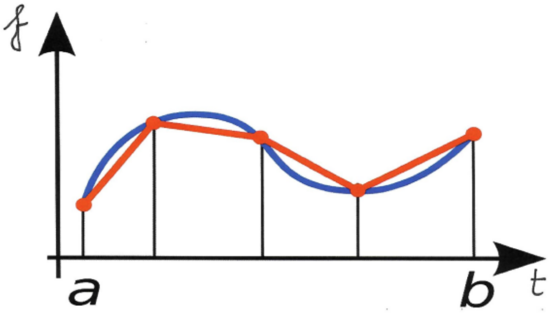


Figure 12: Graphic representation of the Crank-Nicolson method[10]. Again, time is on the horizontal axis, and the function value f is on the vertical axis.

The numerical approximation at time t_{n+1} is given by

$$w_{n+1} = w_n + \frac{1}{2}\Delta t \left(f\left(t_n, w_n\right) + f\left(t_{n+1}, w_{n+1}\right)\right).$$

Since the $f(t_{n+1}, w_{n+1})$-term appears on the right-hand side, we conclude that the Crank-Nicolson method is implicit.

2.4 Example heat equation

In Section 1.2.6, we stated that the Black-Scholes equation is equivalent to the heat equation (see also Appendix A), i.e., if we know the solution to the heat equation, we can derive the solution to the Black-Scholes equation. This also applies when working with an approximated solution instead of an analytical one. Therefore, it is worthwhile to study how we can approximate the solution of the heat equation. This is what we will do in this section, where we make use of [1].

We aim to find a solution $u(x,t)$ of the heat equation, which is given by

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2},$$

for $0 \leq x \leq S_{\max}$ and $0 \leq t \leq T$, subject to

[9] Images taken from https://en.wikipedia.org/wiki/John_Crank and https://en.wikipedia.org/wiki/Phyllis_Nicolson.

[10] Image taken from https://en.wikipedia.org/wiki/File:Composite_trapezoidal_rule_illustration_small.svg.

- the initial condition $u(x,0) = g(x)$, and
- the boundary conditions $u(0,t) = a(t)$ and $u(L,t) = b(t)$.

As we have already mentioned earlier, the space axis is divided into $M+1$ equidistant points, leaving us with the already introduced set

$$X = \{mh\}_{m=0}^{M} = \{0, h, 2h, \ldots, Mh\},$$

where $h = 1/M$. Similarly, the time axis is divided into $N+1$ equidistant points. To avoid confusion with the forward difference (Δ) and backward difference (∇) operator, we will write $k = \Delta t = T/N$. This leads to the set

$$Y = \{nk\}_{n=0}^{N} = \{0, k, 2k, \ldots, Nk = T\}.$$

The grid is formed by points of the form (mh, nk) for $0 \leq m \leq M$ and $0 \leq n \leq N$. We will search for values U_m^n that approximate the solution on the grid, i.e.,

$$U_m^n \approx u(mh, nk),$$

where $0 \leq m \leq M$ and $0 \leq n \leq N$. Figure 13 below gives a sketch of how the grid looks. For this sketch, the time step size appears to be equal to the spatial step size (i.e., $h = k$). This does not necessarily have to be the case!

In Figure 13, the blue dots are known values (these values follow from the initial and boundary conditions). Our goal is to find the values of the red dots. This can be done using finite difference operators to form equations that the grid values U_m^n must satisfy.

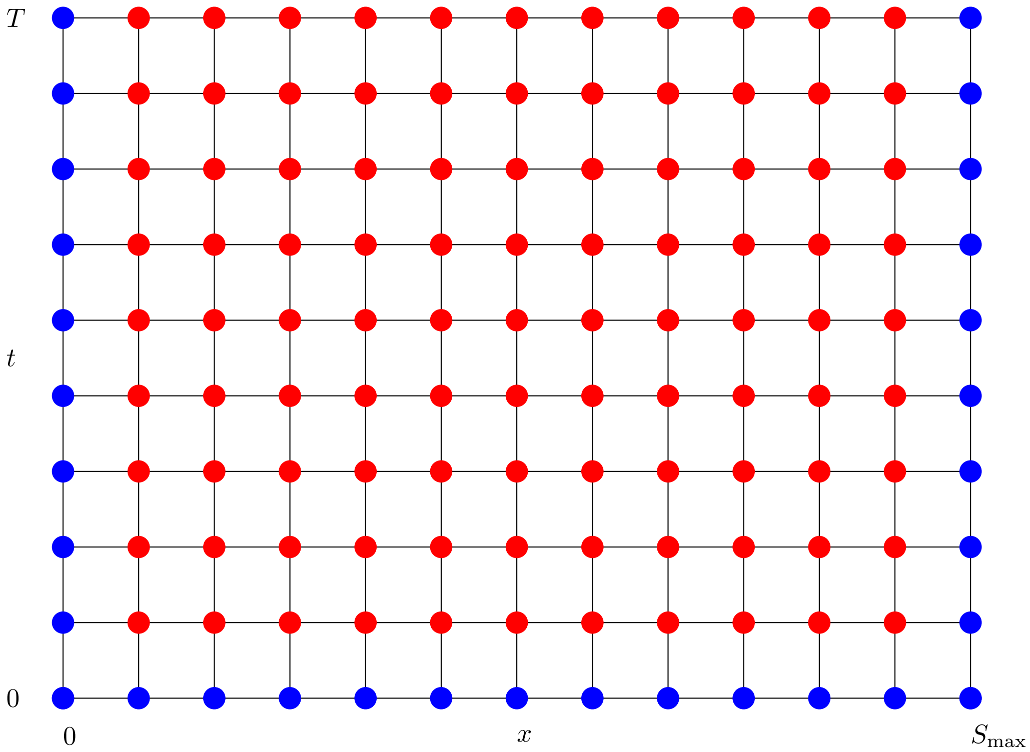


Figure 13: Finite difference grid $\{mh, nk\}_{m,n}$ for $0 \leq m \leq M$ and $0 \leq n \leq N$.

In Figure 13, we observe that all the values of the nodes at time $t = 0$ are known. If, for example, we want to solve the Black-Scholes PDE numerically, it is the other way around: all the values of the nodes at time $t = T$ are known. The methods described in the sections to come will still work because we can also step back in time by introducing the time variable $\tau = T - t$.

2.4.1 Forward difference in time, central difference in space

To solve the heat equation numerically, we can choose to approximate the time derivative $(\partial/\partial t)$ by the forward difference in time, denoted by $k^{-1}\Delta_t$. The second order space derivative $(\partial^2/\partial x^2)$ can be approximated by the second order central difference in space, denoted by $h^{-2}\delta_x^2$. This method is called FTCS (**F**orward difference in **T**ime, **C**entral difference in **S**pace). As a reminder, the differential operators have the following meaning:

$$\begin{aligned}\Delta_t U_m^n &= U_m^{n+1} - U_m^n,\\ \delta_x^2 U_m^n &= U_{m-1}^n - 2U_m^n + U_{m+1}^n.\end{aligned}$$

This leads to the following set of linear equations:

$$\begin{aligned} k^{-1}\Delta_t U_m^n - h^{-2}\delta_x^2 U_m^n &= 0 \\ \frac{U_m^{n+1} - U_m^n}{k} - \frac{U_{m-1}^n - 2U_m^n + U_{m+1}^n}{h^2} &= 0. \end{aligned}$$

By moving the second fraction to the right-hand side, multiplying both sides with k, and then moving the U_m^n to the right-hand side, we obtain

$$U_m^{n+1} = \frac{k}{h^2}U_{m-1}^n + \left(1 - 2\frac{k}{h^2}\right)U_m^n + \frac{k}{h^2}U_{m+1}^n.$$

By introducing $\nu = k/h^2$ as the mesh ratio, we are left with

$$U_m^{n+1} = \nu U_{m-1}^n + (1-2\nu)U_m^n + \nu U_{m+1}^n. \tag{6}$$

If all the approximated solutions at time level n, denoted by $\{U_m^n\}_{m=0}^M$ are known, we can directly compute the approximated solutions at the next time level $n+1$. This is illustrated in Figure 14 and implies that FTCS is an explicit method.

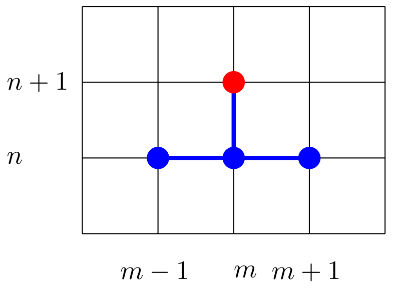


Figure 14: Stencil for FTCS. The grid point values of the blue nodes must be known to compute the grid point value of the red node.

Note that $U_0^{n+1} = a\left((n+1)k\right)$ and $U_M^{n+1} = b\left((n+1)k\right)$ are given by the boundary condition functions a and b, we can use (6) to compute all approximations of the inner values $\{U_m^{n+1}\}_{m=1}^{M-1}$ (the red dots in Figure 13). Together with the fact that all the values $\{U_m^0\}_{m=0}^M$ are known from the initial condition function g, we can compute all the approximations by stepping forward in time.

All of this leads to $M-1$ equalities, given by

$$\begin{array}{rcl} U_1^{n+1} &=& \nu U_0^n + (1-2\nu)U_1^n + \nu U_2^n \\ U_2^{n+1} &=& \nu U_1^n + (1-2\nu)U_2^n + \nu U_3^n \\ \vdots & & \ddots \quad \ddots \quad \ddots \\ U_{M-1}^{n+1} &=& \nu U_{M-2}^n + (1-2\nu)U_{M-1}^n + \nu U_M^n. \end{array}$$

Note that $U_0^n = a(nk)$ and $U_M^n = b(nk)$. By using this notation, we can write this system using matrices and vectors as

$$\mathbf{U}^{n+1} = F\mathbf{U}^n + \mathbf{p}^n, \text{ for } 0 \leq n \leq N-1. \tag{7}$$

Each component in (7) is explained below. We start with the vectors $\mathbf{U}^0$ and $\mathbf{U}^n$.

$$\mathbf{U}^0 = \begin{pmatrix} g(h) \\ g(2h) \\ \vdots \\ \vdots \\ g\left((M-1)h\right) \end{pmatrix} \in \mathbb{R}^{M-1}, \text{ and } \mathbf{U}^n = \begin{pmatrix} U_1^n \\ U_2^n \\ \vdots \\ \vdots \\ U_{M-1}^n \end{pmatrix} \in \mathbb{R}^{M-1}.$$

From (6), it follows that matrix F (from **f**orward time) only has non-zero elements on the main diagonal, subdiagonal, and superdiagonal. Moreover, F is a symmetric matrix:

$$F = \begin{pmatrix} 1-2\nu & \nu & 0 & \cdots & \cdots & 0 \\ \nu & 1-2\nu & \nu & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-2\nu & \nu \\ 0 & \cdots & \cdots & 0 & \nu & 1-2\nu \end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)}.$$

The vector $\mathbf{p}^n$ only has two non-zero elements, namely, the first and the final component (these components correspond to the boundary conditions):

$$\mathbf{p}^n = \begin{pmatrix} \nu a(nk) \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \nu b(nk) \end{pmatrix} \in \mathbb{R}^{M-1}.$$

2.4.2 Backward difference in time, central difference in space

Instead of using forward difference, we can also use the backward difference in time (∇_t) to approximate the time derivative. This method is called BTCS (**B**ackward difference in **T**ime, **C**entral difference in **S**pace). We will perform a similar analysis as in the previous section and start by giving a reminder of the meaning of the following differential operators:

$$\begin{aligned} \nabla_t U_m^n &= U_m^n - U_m^{n-1}, \\ \delta_x^2 U_m^n &= U_{m-1}^n - 2U_m^n + U_{m+1}^n. \end{aligned}$$

This leads to the following set of linear equations:

$$
\begin{aligned}
k^{-1}\nabla_t U_m^n - h^{-2}\delta_x^2 U_m^n &= 0\\
\frac{U_m^n - U_m^{n-1}}{k} - \frac{U_{m-1}^n - 2U_m^n + U_{m+1}^n}{h^2} &= 0.
\end{aligned}
$$

We want to write this as a process that goes from time step n to time step $n+1$, so we increase the time index with 1. Isolating the U_m^n term and using again the abbreviation $\nu = k/h^2$, we obtain

$$-\nu U_{m-1}^{n+1} + (1+2\nu)U_m^{n+1} - \nu U_{m+1}^{n+1} = U_m^n. \tag{8}$$

Unlike (6), we can not compute U^{n+1}-values directly from U^n-values. We first have to solve an equation. This makes BTCS an implicit method.

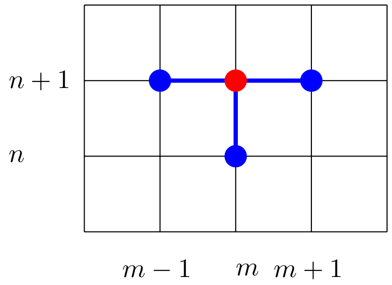


Figure 15: Stencil for BTCS. The grid point values of the blue nodes must be known to compute the grid point value of the red node.

Note that $U_0^{n+1} = a\left((n+1)k\right)$ and $U_M^{n+1} = b\left((n+1)k\right)$ are given by the boundary condition functions a and b, we can use (8) to compute all approximations of the inner values $\{U_m^{n+1}\}_{m=1}^{M-1}$. Since BTCS is an implicit method, this can only be done by solving a system of linear equations, which is given by

$$
\begin{array}{rrrrrrrrl}
-\nu U_0^{n+1}+ & (1+2\nu)U_1^{n+1} & - & \nu U_2^{n+1} & & & & = & U_1^n\\
- & \nu U_1^{n+1} & + & (1+2\nu)U_2^{n+1} & - & \nu U_3^{n+1} & & = & U_2^n\\
 & \ddots & & \ddots & & \ddots & & & \vdots\\
 & & & -\nu U_{M-2}^{n+1} & + & (1+2\nu)U_{M-1}^{n+1} & - \quad \nu U_M^{n+1} & = & U_{M-1}^n.
\end{array}
$$

We perform a similar trick as for the FTCS method by bringing $U_0^{n+1} = a((n+1)k)$ and $U_M^{n+1} = b((n+1)k)$ to the right-hand side to obtain a system of linear equations that has to be solved for $\mathbf{U}^{n+1}$:

$$B\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{q}^n, \text{ for } 0 \le n \le N-1. \tag{9}$$

The vectors $\mathbf{U}^n$ and $\mathbf{U}^{n+1}$ are the same as for the FTCS method. From (8), it follows that matrix B (from **b**ackward time) only has non-zero elements on the main diagonal, subdiagonal, and superdiagonal. Just like the FTCS method, the matrix B for the BTCS method is also symmetric:

$$B = \begin{pmatrix} 1+2\nu & -\nu & 0 & \cdots & \cdots & 0 \\ -\nu & 1+2\nu & -\nu & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1+2\nu & -\nu \\ 0 & \cdots & \cdots & 0 & -\nu & 1+2\nu \end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)}.$$

By the Gershgorin Circle Theorem, all the eigenvalues λ of this matrix are contained in the disc D with center $1+2\nu$ and radius 2ν:

$$D = \{z \in \mathbb{C} : |z - (2\nu+1)| \leq 2\nu\}.$$

Since the matrix is symmetric, all eigenvalues are real, i.e., the imaginary parts of the eigenvalues are 0. Therefore, the eigenvalues are in the interval $[1, 4\nu+1]$. This leads to the conclusion that all the eigenvalues are positive and together with the fact that B is symmetric, we conclude that B is an SPD matrix.

The vector $\mathbf{q}^n$ only has two non-zero elements, namely, the first and the final component (these components correspond to the boundary conditions):

$$\mathbf{q}^n = \begin{pmatrix} \nu a((n+1)k) \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \nu b((n+1)k) \end{pmatrix} \in \mathbb{R}^{M-1}.$$

2.4.3 Crank-Nicolson

When applying the CN method, we make use of a clever trick: we temporarily entertain the idea of an intermediate time level at $(n+\frac{1}{2})k$, i.e., we make use of the half-central difference operator δ. For this choice, the heat equation is approximated by

$$k^{-1}\delta_t U_m^{n+\frac{1}{2}} - h^{-2}\delta_x^2 U_m^{n+\frac{1}{2}} = 0.$$

Since we have introduced points that are not on the grid, we apply the time-averaging operator μ_t on the second term on the left-hand side. By doing this, we obtain

$$k^{-1}\delta_t U_m^{n+\frac{1}{2}} - h^{-2}\delta_x^2 \mu_t U_m^{n+\frac{1}{2}} = 0,$$

leaving us with a scheme containing only points that are actually on the grid:

$$k^{-1}\left(U_m^{n+1} - U_m^n\right) - h^{-2}\delta_x^2 \frac{1}{2}\left(U_m^{n+1} + U_m^n\right) = 0.$$

Using again the mesh ratio $\nu = k/h^2$, we can write the above scheme as

$$2(1+\nu)U_m^{n+1} = \nu U_{m+1}^{n+1} + \nu U_{m-1}^{n+1} + \nu U_{m+1}^{n} + 2(1-\nu)U_m^n + \nu U_{m-1}^n. \tag{10}$$

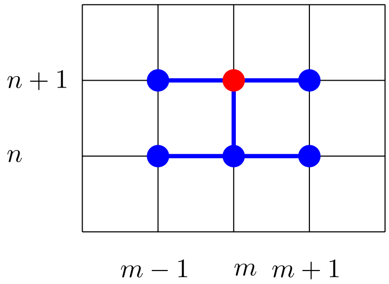


Figure 16: Stencil for Crank-Nicolson. The grid point values of the blue nodes must be known to compute the grid point value of the red node.

By collecting all the boundary value terms into a vector, which we denote by $\mathbf{s}$, we can write (10) in matrix-vector notation.

$$\widehat{B}\mathbf{U}^{n+1} = \widehat{F}\mathbf{U}^n + \mathbf{s}^n, \text{ for } 0 \leq n \leq M-1. \tag{11}$$

The vectors $\mathbf{U}^n$ and $\mathbf{U}^{n+1}$ are the same as for the FTCS and BTCS method. The matrices $\widehat{B}$, $\widehat{F}$ and vector $\mathbf{s}^n$ are given in detail below. By the same reasoning as for matrix B in the BTCS method, we can show that matrix $\widehat{B}$ is SPD.

$$\widehat{B} = \begin{pmatrix} 1+\nu & -\frac{1}{2}\nu & 0 & \cdots & \cdots & 0 \\ -\frac{1}{2}\nu & 1+\nu & -\frac{1}{2}\nu & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{1}{2}\nu & 1+\nu & -\frac{1}{2}\nu \\ 0 & \cdots & \cdots & 0 & -\frac{1}{2}\nu & 1+\nu \end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)},$$

$$\widehat{F} = \begin{pmatrix} 1-\nu & \frac{1}{2}\nu & 0 & \cdots & \cdots & 0 \\ \frac{1}{2}\nu & 1-\nu & \frac{1}{2}\nu & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1}{2}\nu & 1-\nu & \frac{1}{2}\nu \\ 0 & \cdots & \cdots & 0 & \frac{1}{2}\nu & 1-\nu \end{pmatrix} \in \mathbb{R}^{(M-1)\times(M-1)}.$$

The vector $\mathbf{s}^n$ only has two non-zero elements, namely, the first and the final component (these components correspond to a combination of the boundary conditions at time steps n and $n+1$):

$$\mathbf{s}^n = \begin{pmatrix} \frac{1}{2}\nu\left(a(nk)+a((n+1)k)\right) \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{1}{2}\nu\left(b(nk)+b((n+1)k)\right) \end{pmatrix} \in \mathbb{R}^{M-1}.$$

2.5 Local truncation error

2.5.1 Local truncation error for FTCS

If we apply FTCS to the heat equation, we can write out the local accuracy for FTCS:

$$\epsilon_m^n = \frac{\Delta_t u_m^n}{k} - \frac{\delta_x^2 u_m^n}{h^2}.$$

By making use of the Taylor series (1) and (4), we can find the local truncation error:

$$\begin{aligned}\epsilon_m^n &= \frac{k\frac{\partial u}{\partial t} + \frac{1}{2}k^2\frac{\partial^2 u}{\partial t^2} + O\left(k^3\right)}{k} - \frac{h^2\frac{\partial^2 u}{\partial x^2} + \frac{1}{12}h^4\frac{\partial^4 u}{\partial x^4} + O\left(h^6\right)}{h^2} \\ &= \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + \frac{1}{2}k\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^2\right) + O\left(h^4\right).\end{aligned}$$

Since u satisfies the heat equation, we find that

$$\widetilde{\epsilon}_m^n = \frac{1}{2}k\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^2\right) + O\left(h^4\right),$$

which means that FTCS behaves as $O\left(k\right) + O\left(h^2\right)$.

2.5.2 Local truncation error for BTCS

If we apply BTCS to the heat equation, we can write out the local accuracy for BTCS:

$$\epsilon_m^n = \frac{\nabla_t u_m^n}{k} - \frac{\delta_x^2 u_m^n}{h^2}.$$

By making use of the Taylor series (2) and (4), we can find the local truncation error:

$$\begin{aligned}\epsilon_m^n &= \frac{k\frac{\partial u}{\partial t} - \frac{1}{2}k^2\frac{\partial^2 u}{\partial t^2} + O\left(k^3\right)}{k} - \frac{h^2\frac{\partial^2 u}{\partial x^2} + \frac{1}{12}h^4\frac{\partial^4 u}{\partial x^4} + O\left(h^6\right)}{h^2} \\ &= \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{1}{2}k\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^2\right) + O\left(h^4\right).\end{aligned}$$

Since u satisfies the heat equation, we find that

$$\widetilde{\epsilon}_m^n = -\frac{1}{2}k\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^2\right) + O\left(h^4\right),$$

which means that BTCS behaves as $O\left(k\right) + O\left(h^2\right)$, just as FTCS.

2.5.3 Local truncation error for Crank-Nicolson

If we apply Crank-Nicolson to the heat equation, we can write out the local accuracy for CN:

$$\epsilon_m^n = \frac{\delta_t u_m^{n+\frac{1}{2}}}{k} - \frac{\delta_x^2 u_m^{n+\frac{1}{2}}}{h^2}.$$

By making use of the Taylor series (3) and (4), we can find the local truncation error:

$$\begin{aligned}\epsilon_m^n &= \frac{k\frac{\partial u}{\partial t} + \frac{1}{24}k^3\frac{\partial^3 u}{\partial t^3} + O\left(k^4\right)}{k} - \frac{h^2\frac{\partial^2 u}{\partial x^2} + \frac{1}{12}h^4\frac{\partial^4 u}{\partial x^4} + O\left(h^6\right)}{h^2} \\ &= \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} + \frac{1}{24}k^2\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^3\right) + O\left(h^4\right).\end{aligned}$$

Since u satisfies the heat equation, we find that

$$\widetilde{\epsilon}_m^n = \frac{1}{24}k^2\frac{\partial^2 u}{\partial t^2} - \frac{1}{12}h^2\frac{\partial^4 u}{\partial x^4} + O\left(k^3\right) + O\left(h^4\right),$$

which means that CN behaves as $O\left(k^2\right) + O\left(h^2\right)$.

2.6 Numerical Solution Methods

For the BTCS and the CN method, we have to solve a system of linear equations, given by respectively (9) and (11). In this section, we aim to solve a system of linear equations given by $A\mathbf{u} = \mathbf{f}$. Note that (9) and (11) can be written in this form by choosing

- $\mathbf{f} = \mathbf{U}^n + \mathbf{q}^n$ for BTCS, and
- $\mathbf{f} = \widehat{F}\mathbf{U}^n + \mathbf{s}^n$ for CN.

We assume that the matrix A is invertible, i.e., A^{-1} exists. This is a reasonable assumption, since matrices B in (9) and $\widehat{B}$ in (11) are SPD (and thus invertible).

2.6.1 Iterative solution methods

For this master thesis, we will not only use an inverse matrix, but we will also consider iterative solution methods. The name comes from the Latin word ‘iterum’, which means ‘again’. An iterative solution method leads to a sequence of solutions $\{\mathbf{u}^k\}_{k\geq 0}$. In each iteration, the goal is to obtain a new solution that is closer to the real solution $\mathbf{u}$ than the previous solution, i.e.,

$$\|\mathbf{u} - \mathbf{u}^k\| \leq \|\mathbf{u} - \mathbf{u}^{k-1}\|.$$

If the iterative method converges to the real solution, we have that

$$\lim_{k\to\infty} \|\mathbf{u} - \mathbf{u}^k\| = 0,$$

but we can not iterate forever: the algorithm has to stop at some point. We have to stop after a finite number of iterations. The question about when to stop iterating is an amusing one to think about and is answered later in this chapter.

Some interesting background information about iterative solution methods is given in [36]. The idea of using iterative procedures for solving problems is an ancient one. Archimedes' use of the areas of inscribed and circumscribed regular polygons to estimate the area of a circle is a famous instance of an iterative procedure, as is his method of exhaustion for finding the area of a section of a parabola.

2.6.2 Preconditioning

Needless to say, we do not only want our set of approximated solutions $\{\mathbf{u}^k\}_{k\geq 0}$ to converge to the real solution $\mathbf{u}$, we also want this convergence to be as fast as possible. In Theorem 6 that follows in the next section, we will see that the convergence strongly depends on the eigenvalue distribution of matrix A, and, under certain conditions, its condition number $\kappa_p(A)$. The closer this condition number is to 1, the faster convergence tends to be. A **preconditioner** can be used to speed up the convergence. A preconditioner is a matrix M that can be used to describe the set of linear equations equivalently. That is, we go from $A\mathbf{u} = \mathbf{f}$ to the system $M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$. Note that both systems of equations have the same solution. The potential advantage of this method lies in the coefficient matrix $M^{-1}A$, which could have better properties than matrix A. A preconditioning matrix M must satisfy the following three conditions:

- the eigenvalues of $M^{-1}A$ should be clustered around 1, meaning that $\kappa\left(M^{-1}A\right) \ll \kappa(A)$,
- it should be possible to obtain $M^{-1}\mathbf{y}$ at a low cost.

If matrix A is an SPD matrix, we can add another condition, although it can be seen more as a godsend:

- M should be an SPD matrix.

2.7 Examples of Iterative solution methods

In this section, we will consider three iterative solution methods. All three methods are Krylov subspace methods. As a reminder, the subspace

$$K^k = \text{span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\},$$

is called the Krylov-space of dimension k corresponding to matrix A and a vector $\mathbf{v}$. For each of the iterative methods we have that the approximation at iteration k, denoted by $\mathbf{u}^k$ is an element of $K^k(A, \mathbf{r}^0)$, where $\mathbf{r}^0$ is a residual vector that will be explained in more depth later. All three algorithms presented in this section can also be found in [31].

2.7.1 Conjugate Gradient Method

The first iterative solution method that we will use is the Conjugate Gradient method (CG method). Magnus Hestenes and Eduard Stiefel devised this method. They published a paper on this topic in the Journal of Research of the National Bureau of Standards in December 1952 [37].

For this method, we assume that matrix A is symmetric and positive definite. This condition is crucial for deriving the method, which is omitted in this master thesis. The derivation can be found in [31].

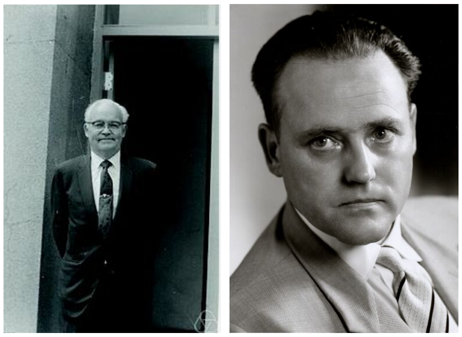


Figure 17: Magnus Hestenes (l) and Eduard Stiefel (r)[11].

Algorithm 1 Conjugate Gradient Method

```
$\mathbf{u}^0 = 0$; $\mathbf{r}^0 = \mathbf{f}$;                                   ▷ initialisation
for $k = 1, 2, \dots$ do                                          ▷ $k$ is the iteration number
    if $k = 1$ then
        $\mathbf{p^1} = \mathbf{r}^0$;
    else if
        then $\beta_k = \frac{\left(\mathbf{r}^{k-1}\right)^\top \mathbf{r}^{k-1}}{\left(\mathbf{r}^{k-2}\right)^\top \mathbf{r}^{k-2}}$;          ▷ $\mathbf{p}^k$ is the search direction vector
        $\mathbf{p}^k = \mathbf{r}^{k-1} + \beta_k \mathbf{p}^{k-1}$;                    ▷ to update $\mathbf{u}^{k-1}$ to $\mathbf{u}^k$
    end if
    $\alpha_k = \frac{\left(\mathbf{r}^{k-1}\right)^\top \mathbf{r}^{k-1}}{\left(\mathbf{p}^k\right)^\top A \mathbf{p}^k}$;
    $\mathbf{u}^k = \mathbf{u}^{k-1} + \alpha_k \mathbf{p}^k$;                                  ▷ update iterate
    $\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_k A \mathbf{p}^k$;                                ▷ update residual
end for
```

The theorem below (cited from [31]) is very powerful because the convergence rate of the CG method is linked to the condition number of matrix A.

Theorem 6. *The iterates* $\mathbf{u}^k$ *obtained from the CG method algorithm satisfy the following inequality:*

$$\|\mathbf{u} - \mathbf{u}^k\|_A \leq 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \|\mathbf{u} - \mathbf{u}^0\|_A. \tag{12}$$

The proof of this theorem is beyond the scope of this master's thesis. Instead, we give an example that serves to demonstrate why the usage of a preconditioning matrix M is powerful. It should be

[11] Images taken from `https://en.wikipedia.org/wiki/Magnus_Hestenes` and `https://en.wikipedia.org/wiki/Eduard_Stiefel`.

mentioned that this example is very artificial since it can be shown that the CG method converges in N steps if A is a $N \times N$ matrix. Since the matrix in Example 2 is 2×2, the CG method would converge in only two steps.

Example 2. *Suppose we want to use the CG method to solve the following system $A\mathbf{u} = \mathbf{f}$, where matrix A is invertible and positive definite:*

$$\begin{pmatrix} 100 & 1 \\ 1 & 10 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 100 \\ 1 \end{pmatrix}.$$

Note that the exact solution is given by $\mathbf{u} = (1, 0)^\top$. It can be shown that $\|A\|_2 = 100.011$ and $\|A^{-1}\|_2 = 0.1$ and thus $\kappa_2(A) = 10.011$. Using (12), we find that

$$\|\mathbf{u} - \mathbf{u}^k\|_A \leq 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \|\mathbf{u} - \mathbf{u}^0\|_A \approx 2 \cdot 0.520^k \|\mathbf{u} - \mathbf{u}^0\|_A.$$

The convergence is acceptable, but it can be much better. To achieve this, write

$$M = \begin{pmatrix} 100 & 0 \\ 0 & 10 \end{pmatrix} \Longrightarrow M^{-1} = \begin{pmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{10} \end{pmatrix}.$$

Note that M is an SPD matrix and for all $\mathbf{y} \in \mathbb{R}^2$, $M^{-1}\mathbf{y}$ can be computed at low cost. We compute $M^{-1}A$:

$$M^{-1}A = \begin{pmatrix} \frac{1}{100} & 0 \\ 0 & \frac{1}{10} \end{pmatrix} \begin{pmatrix} 100 & 1 \\ 1 & 10 \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{100} \\ \frac{1}{10} & 1 \end{pmatrix}.$$

The eigenvalue of this matrix are $\lambda_{1,2} = 1 \pm \frac{1}{100}\sqrt{10}$. It is safe to say that these eigenvalues are indeed clustered around 1. We conclude that M satisfies the three properties of a preconditioning matrix.

The new system $M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}$ (which is equivalent to the original system) now becomes

$$\begin{pmatrix} 1 & \frac{1}{100} \\ \frac{1}{10} & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{10} \end{pmatrix}.$$

Observe that the solution is still given by $\mathbf{u} = (1, 0)^\top$. It can be shown that $\|M^{-1}A\|_2 = 1.056$ and $\|\left(M^{-1}A\right)^{-1}\|_2 = 1.057$, leading to condition number $\kappa_2\left(M^{-1}A\right) \approx 1.116$. Using (12) once more, we obtain

$$\|\mathbf{u} - \mathbf{u}^k\|_A \leq 2 \left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1} \right)^k \|\mathbf{u} - \mathbf{u}^0\|_A \approx 2 \cdot 0.027^k \|\mathbf{u} - \mathbf{u}^0\|_A.$$

Now, the upper bound for the distance between $\mathbf{u}$ and $\mathbf{u}^k$ is much sharper than before.

2.7.2 Bi-Conjugate gradient stabilized method

In the previous section, we have seen that the CG method can only be used if matrix A is SPD. This is not always the case. Therefore, mathematicians were also looking for methods that could be used for general systems of linear equations. One such method is the Bi-Conjugate Gradient type method. We have short recurrences, which means that only the results of one foregoing step are necessary; work and memory do not increase for an increasing number of iterations. Unfortunately, we do not have an optimality property. The algorithm below makes use of a preconditioner matrix M.

The inventor of the Bi-Conjugate gradient stabilized method (abbreviated as the Bi-CGSTAB method) is a Dutch mathematician named Henk van der Vorst. In his paper [38], published in 1992, he states that convergence for Bi-CGSTAB is faster and smoother than for a regular Bi-CG method.

Figure 18: Henk van der Vorst[12].

[12] Image taken from `https://elbd.sites.uu.nl/2020/01/09/henk-van-der-vorst/`.

Algorithm 2 **Bi-CGSTAB method**

```
$\mathbf{u}^0$ is an initial guess; $\mathbf{r}^0 = \mathbf{f} - A\mathbf{u}^0$;
$\bar{\mathbf{r}}^0$ is an arbitrary vector, such that $\left(\bar{\mathbf{r}}^0\right)^\top \mathbf{r}^0 \neq 0$, e.g., $\bar{\mathbf{r}}^0 = \mathbf{r}^0$;
$\rho_{-1} = \alpha_{-1} = \omega_{-1} = 1$;
$\mathbf{v}^{-1} = \mathbf{p}^{-1} = 0$;
for $i = 1, 2, \dots$ do                                    ▷ $i$ is the iteration number
    $\rho_i = \left(\bar{\mathbf{r}}^0\right)^\top \mathbf{r}^i$; $\beta_{i-1} = (\rho_i/\rho_{i-1})(\alpha_{i-1}/\omega_{i-1})$;
    $\mathbf{p}^i = \mathbf{r}^i + \beta_{i-1}\left(\mathbf{p}^{i-1} - \omega_{i-1}\mathbf{v}^{i-1}\right)$;
    $\hat{\mathbf{p}} = M^{-1}\mathbf{p}^i$;
    $\mathbf{v}^i = A\hat{\mathbf{p}}$;
    $\alpha_i = \rho_i / \left(\bar{\mathbf{r}}^0\right)^\top \mathbf{v}^i$;
    $\mathbf{s} = \mathbf{r}^i - \alpha_i \mathbf{v}^i$;
    if $\|s\|$ is small enough then
        $\mathbf{u}^{i+1} = \mathbf{u}^i + \alpha_i \hat{\mathbf{p}}$; quit;
    end if
    $\mathbf{z} = M^{-1}\mathbf{s}$;
    $\mathbf{t} = A\mathbf{z}$;
    $\omega_i = \mathbf{t}^\top \mathbf{s} / \mathbf{t}^\top \mathbf{t}$;
    $\mathbf{u}^{i+1} = \mathbf{u}^i + \alpha_i \hat{\mathbf{p}} + \omega_i \mathbf{z}$;
    if $\mathbf{u}^{i+1}$ is accurate enough then
        quit;
    end if
    $\mathbf{r}^{i+1} = \mathbf{s} - \omega_i \mathbf{t}$;
end for
```

2.7.3 General Minimal Residual Method

The GMRES method [39] was developed in 1986 by Yousef Saad and Martin H. Schultz and stems from the MINRES method [40], which was brought to life eleven years earlier. This type of method features long recurrences, i.e., the amount of work per iteration, and the required memory grows for an increasing number of iterations. An advantage of this method over the Bi-CGSTAB method is that the GMRES-type methods do have an optimality property.

Figure 19: Yousef Saad (l) and Martin H. Schultz (r)[13].

Algorithm 3 GMRES method

```
Choose $\mathbf{u}^0$ and compute $\mathbf{r}^0 = \mathbf{f} - A\mathbf{u}^0$ and $\mathbf{v}^1 = \mathbf{r^0}/\|\mathbf{r}^0\|_2$;
for $j = 1, \dots, k$ do
    $\mathbf{v}^{j+1} = A\mathbf{v}^j$;
    for $i = 1, \dots, j$ do
        $h_{ij} := \left(\mathbf{v}^{j+1}\right)^\top \mathbf{v}^i$; $\mathbf{v}^{j+1} := \mathbf{v}^{j+1} - h_{ij}\mathbf{v}^i$;
    end for
    $h_{j+1,j} := \|\mathbf{v}^{j+1}\|_2$; $\mathbf{v}^{j+1} := \mathbf{v}^{j+1}/h_{j+1,j}$
end for
The entries of the upper $k+1 \times k$ Hessenberg matrix $\bar{H}_k$ are the scalars $h_{ij}$.
```

In GMRES the approximate solution $\mathbf{u}^k = \mathbf{u}^0 + \mathbf{z}^k$ with $\mathbf{z}^k \in K^k\left(A; \mathbf{r}^0\right)$ is such that

$$\|\mathbf{r}^k\|_2 = \|\mathbf{f} - A\mathbf{u}^k\|_2 = \min_{\mathbf{z} \in K^k(A;\mathbf{r}^0)} \|\mathbf{r}^0 - A\mathbf{z}\|_2.$$

2.8 Stopping criteria

The three algorithms which are presented in the previous section do not contain clear stopping criteria. That is because we can make a choice. In [31], three stopping criteria are given. Some are better than others as we will see.

- $\|\mathbf{r}^k\| \leq \epsilon$. This stopping criterium is not scaling invariant. Therefore, scaling needs to be introduced.
- $\|\mathbf{r}^k\|/\|\mathbf{r}^0\| \leq \epsilon$. The main problem with this stopping criteria is that the required accuracy increases with the quality of the initial guess.

[13] Images taken from `https://www-users.cse.umn.edu/~saad/` and `https://nl.wikipedia.org/wiki/Martin_Schulz`.

- Given that the two stopping criteria above have major disadvantages, the most reliable stopping criteria given by $\|\mathbf{r}^k\|/\|\mathbf{f}\| \leq \epsilon$. This stopping criterion is both scaling invariant and does not increase with the quality of the initial guess.

3 Stability Analysis of the Numerical Methods

3.1 Amplification factor

Given the content in Chapters 2.3.2 and 2.3.3 about explicit and implicit methods, one could be tempted to say that using an explicit method is more straightforward than an implicit method. In particular, applying an explicit method does not lead to an equation that has to be solved for either w_{n+1} or $\mathbf{U}^{n+1}$. Although this is true, there are some strong arguments in favor of implicit methods. One of these arguments has to do with stability. According to [33], a physical phenomenon is called stable if a small perturbation of the parameters (including the initial condition) results in a small difference in the solution.

We perform the stability analysis for the three time-stepping methods mentioned in Chapter 2.3 (namely, Forward Euler, Backward Euler, and Crank-Nicolson). The PDE that we will use is the so-called test equation, which, for $\lambda \in \mathbb{R}$, is given by

$$\begin{cases} y' = \lambda y + g(t), \ t > t_0, \\ y\left(t_0\right) = y_0. \end{cases} \tag{13}$$

The problem might get perturbed due to, for example, rounding errors. The perturbed system looks similar to the original system. It is given by

$$\begin{cases} \tilde{y}' = \lambda \tilde{y} + g(t), \ t > t_0, \\ \tilde{y}\left(t_0\right) = y_0 + \epsilon_0, \end{cases} \tag{14}$$

where ϵ_0 is the error at time $t = 0$. Subtracting (13) from (14) gives a differential equation for the error ϵ:

$$\begin{cases} \epsilon' = \tilde{y}' - y' = \lambda\left(\tilde{y} - y\right) = \lambda \epsilon, \ t > t_0, \\ \epsilon\left(t_0\right) = \epsilon_0, \end{cases} \tag{15}$$

with a solution given by

$$\epsilon(t) = \epsilon_0 e^{\lambda(t-t_0)}.$$

Of course, we want ϵ to be as small as possible. From this viewpoint, the following definition arises naturally.

Definition 14. *An initial value problem is called stable if*

$$|\epsilon(t)| < \infty \textit{ for all } t > t_0,$$

and absolutely stable if it is stable and

$$\lim_{t \to \infty} |\epsilon(t)| = 0.$$

If $|\epsilon(t)|$ is unbounded, then the initial-value problem is unstable.

Since

$$\lim_{t\to\infty} \epsilon(t) = \begin{cases} 0 & \text{if } \lambda < 0, \\ \epsilon_0 & \text{if } \lambda = 0, \\ \infty & \text{if } \lambda > 0. \end{cases}$$

We conclude that (15) is stable if $\lambda \leq 0$ and strictly stable if $\lambda < 0$.

We have already stated that the numerical approximation for y_{n+1} is denoted by w_{n+1}. Next to this, we denote numerical approximation for the perturbed problem by $\tilde{w}_{n+1}$. The numerical approximation for the error is then given by $\tilde{\epsilon}_n = w_n - \tilde{w}_n$. The next definition is again cited from [33].

Definition 15 (**Amplification factor**). *Given the test equation and a time-stepping method, we can find a recursive formula for the error:*

$$\tilde{\epsilon}_{n+1} = Q\left(\lambda \Delta t\right) \tilde{\epsilon}_n.$$

The value of $Q\left(\lambda\Delta t\right)$ determines the factor by which the existing perturbation is amplified. Therefore, this factor is called the amplification factor.

Theorem 7 (**Stability and amplification factor**). *A time-stepping method is*

$$\begin{aligned} \textit{absolutely stable} &\iff \left|Q\left(\lambda\Delta t\right)\right| < 1, \\ \textit{stable} &\iff \left|Q\left(\lambda\Delta t\right)\right| \leq 1, \\ \textit{unstable} &\iff \left|Q\left(\lambda\Delta t\right)\right| > 1. \end{aligned}$$

If we assume that $\lambda < 0$, it turns out that the above-mentioned implicit methods (Backward Euler and Crank-Nicolson) have much nicer properties regarding stability than the explicit method (Forward Euler).

Theorem 8. *Consider the test equation* (13) *and assume that $\lambda < 0$. Then, the Forward Euler method is*

$$\begin{aligned} \textit{stable} &\iff \Delta t \leq -\frac{2}{\lambda}, \\ \textit{absolutely stable} &\iff \Delta t < -\frac{2}{\lambda}. \end{aligned}$$

Proof. For the Forward Euler method, we can write the numerical approximation w_{n+1} in terms of w_n:

$$w_{n+1} = w_n + \Delta t \left(\lambda w_n + g\left(t_n\right)\right). \tag{16}$$

For the perturbed initial-value problem, we obtain:

$$\tilde{w}_{n+1} = \tilde{w}_n + \Delta t \left(\lambda \tilde{w}_n + g\left(t_n\right)\right). \tag{17}$$

Using that $\tilde{\epsilon}_n = w_n - \tilde{w}_n$, we can subtract (17) from (16). By doing this, we obtain:

$$\begin{aligned}\tilde{\epsilon}_{n+1} &= w_{n+1} - \tilde{w}_{n+1}\\ &= w_n + \Delta t\left(\lambda w_n + g\left(t_n\right)\right) - \left(\tilde{w}_n + \Delta t\left(\lambda \tilde{w}_n + g\left(t_n\right)\right)\right)\\ &= \left(1 + \lambda \Delta t\right)\left(w_n - \tilde{w}_n\right)\\ &= (1 + \lambda \Delta t)\tilde{\epsilon}_n.\end{aligned}$$

From here, it follows that $Q\left(\lambda \Delta t\right) = 1 + \lambda \Delta t$. By Theorem 7, we need to have that

$$-1 \leq 1 + \lambda \Delta t \leq 1,$$

which comes down to

$$-2 \leq \lambda \Delta t \leq 0.$$

The right inequality is automatically satisfied, since $\Delta t > 0$ and $\lambda < 0$. Dividing both sides by λ leads to

$$\Delta t \leq -\frac{2}{\lambda},$$

which finishes the proof. □

Theorem 9. *Consider the test equation* (13) *and assume that* $\lambda < 0$*. Then, both Backward Euler and Crank-Nicolson are unconditionally stable, i.e., both methods are stable no matter the value of* Δt*.*

Proof. We start with the Backward Euler method.

For this method, we will write the numerical approximation w_{n+1} in terms of w_n:

$$w_{n+1} = w_n + \Delta t\left(\lambda w_{n+1} + g\left(t_{n+1}\right)\right). \tag{18}$$

For the perturbed initial-value problem, we obtain:

$$\tilde{w}_{n+1} = \tilde{w}_n + \Delta t\left(\lambda \tilde{w}_{n+1} + g\left(t_{n+1}\right)\right). \tag{19}$$

Using that $\tilde{\epsilon}_n = w_n - \tilde{w}_n$, we can subtract (19) from (18). By doing this, we obtain:

$$\begin{aligned}\tilde{\epsilon}_{n+1} &= w_{n+1} - \tilde{w}_{n+1}\\ &= w_n + \Delta t\left(\lambda w_{n+1} + g\left(t_{n+1}\right)\right) - \left(\tilde{w}_n + \Delta t\left(\lambda \tilde{w}_{n+1} + g\left(t_{n+1}\right)\right)\right)\\ &= w_n - \tilde{w}_n + \lambda \Delta t\left(w_{n+1} - \tilde{w}_{n+1}\right)\\ &= \tilde{\epsilon}_n + \lambda \Delta t \tilde{\epsilon}_{n+1}.\end{aligned}$$

From here, we can determine the amplification factor:

$$\tilde{\epsilon}_{n+1} = \frac{1}{1 - \lambda \Delta t}\tilde{\epsilon}_n \Longrightarrow Q\left(\lambda \Delta t\right) = \frac{1}{1 - \lambda \Delta t}.$$

By Theorem 7, we need to have that

$$-1 \leq \frac{1}{1-\lambda\Delta t} \leq 1.$$

Since $\Delta t > 0$ and $\lambda < 0$, it follows that $1 - \lambda\Delta t > 1$, so this condition holds for every Δt. We conclude that the Backward Euler method is unconditionally stable for $\lambda < 0$.

For the Crank-Nicolson method, a similar analysis will be performed. We will again write the numerical approximation w_{n+1} in terms of w_n:

$$w_{n+1} = w_n + \frac{1}{2}\Delta t \left(\lambda w_n + g\left(t_n\right) + \lambda w_{n+1} + g\left(t_{n+1}\right)\right). \tag{20}$$

For the perturbed initial-value problem, we obtain:

$$\tilde{w}_{n+1} = \tilde{w}_n + \frac{1}{2}\Delta t \left(\lambda \tilde{w}_n + g\left(t_n\right) + \lambda \tilde{w}_{n+1} + g\left(t_{n+1}\right)\right). \tag{21}$$

Using that $\tilde{\epsilon}_n = w_n - \tilde{w}_n$, we can subtract (21) from (20). By doing this, we obtain:

$$\begin{aligned}
\tilde{\epsilon}_{n+1} &= w_{n+1} - \tilde{w}_{n+1} \\
&= w_n + \frac{1}{2}\Delta t \left(\lambda w_n + g\left(t_n\right) + \lambda w_{n+1} + g\left(t_{n+1}\right)\right) - \left(\tilde{w}_n + \frac{1}{2}\Delta t \left(\lambda \tilde{w}_n + g\left(t_n\right) + \lambda \tilde{w}_{n+1} + g\left(t_{n+1}\right)\right)\right) \\
&= \left(1 + \frac{1}{2}\lambda\Delta t\right)\left(w_n - \tilde{w}_n\right) + \frac{1}{2}\lambda\Delta t \left(w_{n+1} - \tilde{w}_{n+1}\right) \\
&= \left(1 + \frac{1}{2}\lambda\Delta t\right)\tilde{\epsilon}_n + \frac{1}{2}\lambda\Delta t \tilde{\epsilon}_{n+1}.
\end{aligned}$$

From here, we can determine the amplification factor:

$$\tilde{\epsilon}_{n+1} = \frac{1 + \frac{1}{2}\lambda\Delta t}{1 - \frac{1}{2}\lambda\Delta t}\tilde{\epsilon}_n \Longrightarrow Q\left(\lambda\Delta t\right) = \frac{1 + \frac{1}{2}\lambda\Delta t}{1 - \frac{1}{2}\lambda\Delta t}.$$

By Theorem 7, we need to have that

$$-1 \leq \frac{1 + \frac{1}{2}\lambda\Delta t}{1 - \frac{1}{2}\lambda\Delta t} \leq 1.$$

Since $\Delta t > 0$ and $\lambda < 0$, it follows that $|1 - \frac{1}{2}\lambda\Delta t| > |1 + \frac{1}{2}\lambda\Delta t|$, so the condition above holds for every Δt. We can conclude that the Crank-Nicolson method is also unconditionally stable for $\lambda < 0$. This finishes the proof. □

3.2 Von Neumann stability

This section is partly cited from from [1]. During the mid-twentieth century, John von Neumann [41], a Hungarian mathematician who also had great knowledge about physics, astronomy, and biology, worked among others on a method to determine the stability of a finite difference scheme [42]. This method makes use of Fourier analysis. We will not dive into this topic, but the definition of stability in the sense of von Neumann is stated below and is understandable without much knowledge about Fourier analysis. Note that a Von Neumann analysis can only be performed if the partial differential equation is linear. For this master's project, this will always be the case.

Figure 20: John von Neumann[14].

Definition 16 (Von Neumann stability). *Let* $\mathbf{i}$ *be the unit imaginary number:* $\mathbf{i} = \sqrt{-1}$*. A finite difference method generating approximation* U_m^n *is stable in the sense of von Neumann if, ignoring initial and boundary conditions, under the substitution*

$$U_m^n = \xi^n e^{\mathbf{i}\beta m h},$$

it follows that $|\xi| \leq 1$ *for all* $\beta h \in [-\pi, \pi]$*.*

In the Chapter 2.4.1, we have introduced the mesh ratio $\nu = k/h^2$. This constant is important for determining if a finite difference method is stable in the sense of von Neumann. This will be emphasized in two theorems in the next section. These theorems will be stated with proof.

3.3 Stability analysis of the heat equation

In Chapters 2.4.1 and 2.4.2, we have introduced the FTCS and BTCS methods for discretizing the heat equation (see Appendix A). We perform a stability analysis in the sense of von Neumann for both methods in the proof of the following two theorems. This analysis is interesting since the heat equation is equivalent to the Black-Scholes equation. Moreover, the PDE in the green bond model (see Chapter 4.1) has some similarities with the Black-Scholes equation.

Theorem 10. *The stability in the sense of von Neumann for FTCS is equivalent to*

$$\nu \leq \frac{1}{2}.$$

Proof. As a reminder, the scheme for FTCS is given by

$$U_m^{n+1} = \nu U_{m-1}^n + (1 - 2\nu) U_m^n + \nu U_{m+1}^n.$$

[14] Image taken from `https://www.econlib.org/library/Enc/bios/Neumann.html`.

By substituting $U_m^n = \xi^n e^{\mathbf{i}\beta m h}$, we obtain

$$\xi^{n+1} e^{\mathbf{i}\beta m h} = \nu \xi^n e^{\mathbf{i}\beta (m-1) h} + (1-2\nu)\xi^n e^{\mathbf{i}\beta m h} + \nu \xi^n e^{\mathbf{i}\beta (m+1) h}.$$

Multiplying both sides with $\xi^{-n} e^{-\mathbf{i}\beta m h}$ ensures that we are left with ξ on the left-hand side.

$$\begin{aligned}\xi &= \nu e^{-\mathbf{i}\beta h} + (1-2\nu) + \nu e^{\mathbf{i}\beta h} \\ &= 1 + \nu \left(e^{-\mathbf{i}\beta h} - 2 + e^{\mathbf{i}\beta h}\right) \\ &= 1 + \nu \left(e^{\mathbf{i}\frac{1}{2}\beta h} - e^{-\mathbf{i}\frac{1}{2}\beta h}\right)^2 .\end{aligned}$$

We proceed by using that $e^{\mathbf{i}x} = \cos(x) + \mathbf{i}\sin(x)$, $\cos(-x) = \cos(x)$, and $\sin(-x) = -\sin(x)$. This gives

$$\begin{aligned}\xi &= 1 + \nu \left(\cos\left(\frac{1}{2}\beta h\right) + \mathbf{i}\sin\left(\frac{1}{2}\beta h\right) - \left(\cos\left(-\frac{1}{2}\beta h\right) + \mathbf{i}\sin\left(-\frac{1}{2}\beta h\right)\right)\right)^2 \\ &= 1 + \nu \left(2\mathbf{i}\sin\left(\frac{1}{2}\beta h\right)\right)^2 \\ &= 1 - 4\nu \sin^2\left(\frac{1}{2}\beta h\right).\end{aligned}$$

We showed that

$$\left(e^{\mathbf{i}\beta h} - 2 + e^{-\mathbf{i}\beta h}\right) = -4\sin^2\left(\frac{1}{2}\beta h\right), \tag{22}$$

and we will use this result multiple times in the proofs to come.

The condition $|\xi| \leq 1$ holds true if and only if

$$-1 \leq 1 - 4\nu \sin^2\left(\frac{1}{2}\beta h\right) \leq 1,$$

which can be simplified to

$$0 \leq \nu \sin^2\left(\frac{1}{2}\beta h\right) \leq \frac{1}{2}.$$

For $\beta h \in [-\pi, \pi]$, the squared sine term takes value in $[0,1]$. Therefore, the stability in the sense of von Neumann for FTCS is equivalent to

$$\nu \leq \frac{1}{2}.$$

This finishes the proof. □

The above theorem implies that we should be careful when choosing values for k and h if we use the FTCS method. We should never forget to check if the mesh ratio satisfies the inequality $\nu \leq \frac{1}{2}$. BTCS is an implicit method, i.e., it requires solving an equation to make a step forward in time. This might sound like more work has to be done. Though this is true, an advantage over FTCS is stated in the next theorem.

Theorem 11. *The BTCS method is unconditionally stable in the sense of von Neumann, that is, stability in the sense of von Neumann is satisfied for all $\nu > 0$.*

Proof. As a reminder, the scheme for BTCS is given by

$$-\nu U_{m-1}^{n+1} + (1+2\nu)U_m^{n+1} - \nu U_{m+1}^{n+1} = U_m^n.$$

By substituting $U_m^n = \xi^n e^{\mathbf{i}\beta mh}$, we obtain

$$-\nu\xi^{n+1}e^{\mathbf{i}\beta(m-1)h} + (1+2\nu)\xi^{n+1}e^{\mathbf{i}\beta mh} - \nu\xi^{n+1}e^{-\mathbf{i}\beta(m+1)h} = \xi^n e^{\mathbf{i}\beta mh}.$$

Multiplying both sides with $\xi^{-n}e^{-\mathbf{i}\beta mh}$ gives the following:

$$\begin{aligned} -\nu\xi e^{-\mathbf{i}\beta h} + (1+2\nu)\xi - \nu\xi e^{\mathbf{i}\beta h} &= 1 \\ \xi - \xi\nu\left(e^{\mathbf{i}\beta h} - 2 + e^{-\mathbf{i}\beta h}\right) &= 1. \end{aligned}$$

We use equation (22) to eliminate the imaginary terms for the next step.

$$\begin{aligned} \xi - \xi\nu\left(-4\sin^2\left(\frac{1}{2}\beta h\right)\right) &= 1 \\ \xi\left(1 + 4\nu\sin^2\left(\frac{1}{2}\beta h\right)\right) &= 1. \end{aligned}$$

From here we can derive an expression for ξ:

$$\xi = \frac{1}{1 + 4\nu\sin^2\left(\frac{1}{2}\beta h\right)}. \tag{23}$$

Since $\nu > 0$ and $\sin^2\left(\frac{1}{2}\beta h\right) \in [0,1]$ for $\beta h \in [-\pi, \pi]$, we find that $\xi > 0$. Moreover, the numerator of (23) is always smaller than or equal to the denominator. By this reasoning, we can conclude that $|\xi| \leq 1$, meaning that BTCS is unconditionally stable in the sense of von Neumann. This ends the proof. □

Like BTCS, the Crank-Nicolson method is also implicit. The following theorem demonstrates that there are more similarities between CN and BTCS.

Theorem 12. *The Crank-Nicolson method is unconditionally stable in the sense of von Neumann, that is, stability in the sense of von Neumann is satisfied for all $\nu > 0$.*

Proof. As a reminder, the scheme for CN is given by

$$2(1+\nu)U_m^{n+1} = \nu U_{m+1}^{n+1} + \nu U_{m-1}^{n+1} + \nu U_{m+1}^n + 2(1-\nu)U_m^n + \nu U_{m-1}^n.$$

By substituting $U_m^n = \xi^n e^{\mathbf{i}\beta mh}$, we obtain

$$\begin{aligned} 2(1+\nu)\xi^{n+1}e^{\mathbf{i}\beta mh} = {} & \nu\xi^{n+1}e^{\mathbf{i}\beta(m+1)h} + \nu\xi^{n+1}e^{\mathbf{i}\beta(m-1)h} + \nu\xi^n e^{\mathbf{i}\beta(m+1)h} \\ & + 2(1-\nu)\xi^n e^{\mathbf{i}\beta mh} + \nu\xi^n e^{\mathbf{i}\beta(m-1)h}. \end{aligned}$$

Next, we multiply both sides with $\xi^{-n}e^{-\mathbf{i}\beta mh}$. This leaves us with

$$2(1+\nu)\xi = \nu\xi e^{\mathbf{i}\beta h} + \nu\xi e^{-\mathbf{i}\beta h} + \nu e^{\mathbf{i}\beta h} + 2(1-\nu) + \nu e^{-\mathbf{i}\beta h},$$

and collecting the ξ-terms on the left-hand side results in the following expression:

$$\xi\left(2-\nu\left(e^{\mathbf{i}\beta h}-2+e^{-\mathbf{i}\beta h}\right)\right)=\nu\left(e^{\mathbf{i}\beta h}-2+e^{-\mathbf{i}\beta h}\right)+2.$$

We will use (22) both on the left-hand and right-hand side:

$$\xi\left(2+4\nu\sin^2\left(\frac{1}{2}\beta h\right)\right)=2-4\nu\sin^2\left(\frac{1}{2}\beta h\right).$$

From here, we can find an expression for ξ.

$$\xi=\frac{2-4\nu\sin^2\left(\frac{1}{2}\beta h\right)}{2+4\nu\sin^2\left(\frac{1}{2}\beta h\right)}.$$

Since $\nu>0$ and $\sin^2\left(\frac{1}{2}\beta h\right)\in[0,1]$ for $\beta h\in[-\pi,\pi]$, we have that

$$\left|2-4\nu\sin^2\left(\frac{1}{2}\beta h\right)\right|\leq\left|2+4\nu\sin^2\left(\frac{1}{2}\beta h\right)\right|,$$

and thus we find that $|\xi|\leq 1$ for all $\nu>0$. We conclude that the Crank-Nicolson method is unconditionally stable in the sense of von Neumann, which finishes the proof. □

4 Stability analysis for the green bond

In Chapter 3, we obtained results on the stability of FTCS, BTCS, and the Crank-Nicolson method for the heat equation. These results are important since it can be shown that the heat equation is equivalent to the Black-Scholes equation (see Appendix A).

4.1 The green bond model

The PDE for finding the value of a green bond, also called a coupon value, based on the interest rate and the carbon price is derived by Juriaan Rutten [29]. The goal of his master's project was to solve this PDE on the domain

$$D = \{(t, r, c) \mid 0 \leq t \leq T,\ r_{\min} \leq r_t \leq r_{\max},\ c_{\min} \leq c_t \leq c_{\max}\}.$$

The PDE itself is given by:

$$\begin{aligned}\frac{\partial V}{\partial t} &+ (\mu c - \lambda_c \sigma_c c)\frac{\partial V}{\partial c} + \left(\alpha(\beta - r) - \lambda_r \sigma_r \sqrt{r}\right)\frac{\partial V}{\partial r} \\ &+ \frac{1}{2}\left(\sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} + \sigma_r^2 r \frac{\partial^2 V}{\partial r^2} + 2c\rho\sigma_c\sigma_r\sqrt{r}\frac{\partial^2 V}{\partial r \partial c}\right) - rV = 0. \end{aligned} \tag{24}$$

The boundary conditions are as follows:

$$\begin{cases} V(T, r, c) = (c_T - K)^+, \\ V(t, r, c_{\min}) = 0, \\ V(t, r, c_{\max}) = \left(c_{\max} - Ke^{-r(T-t)}\right)^+, \\ V(t, r_{\min}, c) = h_1(t, c), \\ V(t, r_{\max}, c) = h_2(t, c), \end{cases}$$

where h_1 can be obtained by solving the PDE below

$$\begin{cases} \frac{\partial V}{\partial t} + (\mu c - \lambda_c \sigma_c c)\frac{\partial V}{\partial c} + \frac{1}{2}\sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} - r_{\min} V = 0, \\ V(t, r_{\min}, c_{\max}) = 0, \\ V(t, r_{\min}, c_{\max}) = \left(c_{\max} - Ke^{-r_{\min}(T-t)}\right)^+, \\ V = (T, r_{\min}, c) = (c_T - K)^+. \end{cases}$$

Similarly, h_2 can be found by solving the following PDE:

$$\begin{cases} \frac{\partial V}{\partial t} + (\mu c - \lambda_c \sigma_c c)\frac{\partial V}{\partial c} + \frac{1}{2}\sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} - r_{\max} V = 0, \\ V(t, r_{\max}, c_{\min}) = 0, \\ V(t, r_{\max}, c_{\max}) = \left(c_{\max} - Ke^{-r_{\max}(T-t)}\right)^+, \\ V = (T, r_{\max}, c) = (c_T - K)^+. \end{cases}$$

This partial differential equation looks somewhat similar to the Black-Scholes equation, but this model also considers the carbon price c. Moreover, since we have a factor $\sqrt{r}$, this model only allows non-negative values for the risk-free rate r. This PDE depends on several parameters, the most important ones are listed below. For further details, such as the derivation of the model, please consult Chapter 2 of [29].

- V: the value of the bond,
- c: the carbon price,
- r: the risk-free interest rate,
- λ_c, λ_r: the market prices of risk for the carbon price and the risk-free rate, and
- σ_c, σ_r: the volatility of the carbon price and the risk-free rate.

4.2 Methodology

It is more difficult to perform a stability analysis for (24) than for the heat equation in Chapter 2.4. The main reason for this fact is that (24) contains an extra dimension: the carbon price. To go easy on ourselves, we split up the problem into subproblems, which we will solve one by one.

- In Section 4.3, we will assume that the carbon price c and the risk-free rate r are both constant.
- In Section 4.4, we will assume that c is constant, while r is not.
- In Section 4.5, we assume r to be constant, while c is not.
- In Section 4.6, we assume both r and c to be changing over time.

The first problem is relatively easy since we can find an analytical solution for the PDE. The other problems are more difficult and require much more attention. We will use both forward and backward differences for the first-order partial derivatives $\frac{\partial V}{\partial c}$ and $\frac{\partial V}{\partial r}$. For the second order partial derivatives $\frac{\partial^2 V}{\partial c^2}$ and $\frac{\partial^2 V}{\partial r^2}$ we will use central differences.

It is already mentioned that, unlike the initial condition for the heat equation, we have a final condition for (24). After discretizing the grid, we want to compute the values of the nodes corresponding to time $t = 0$. This means we take steps backward in time to obtain these values. To obtain the values of the nodes in row n, corresponding to time t_n, we make use of the fact that we know the values of the nodes in row $n + 1$, corresponding to time t_{n+1}. For this reason, forward differences will lead to implicit methods and backward differences will lead to explicit methods.

4.3 Part 0: the interest rate r and the carbon price c are both constant

We assume that $r \in [0, 1]$ (corresponding to 0% and 100%) and $c \in \mathbb{R}_{>0}$ are both fixed. For this choice, we have that

$$\frac{\partial V}{\partial c} = \frac{\partial V}{\partial r} = \frac{\partial^2 V}{\partial c^2} = \frac{\partial^2 V}{\partial r^2} = \frac{\partial^2 V}{\partial r \partial c} = 0.$$

Substituting these values into the PDE leaves us with

$$\frac{\partial V}{\partial t} - rV = 0.$$

For an initial value condition $V(t_0) = v_0$, this differential equation looks very similar to the test equation (where $\lambda = r > 0$ and $g \equiv 0$).

$$\begin{cases} \frac{\partial V}{\partial t} = rV, t > t_0, \\ V(t_0) = v_0 \end{cases} \implies V(t) = v_0 e^{rt}.$$

For $r = 0$ we find that V itself is constant. For $0 < r \leq 1$, all the time stepping methods are not stable: since $\Delta t > 0$ and $r \in (0,1)$, we have

- Forward Euler: $Q(r\Delta t) = 1 + r\Delta t > 1$,
- Backward Euler: $Q(r\Delta t) = \frac{1}{1 - r\Delta t} > 1$, and
- Crank-Nicolson: $Q(r\Delta t) = \frac{1 + \frac{1}{2} r\Delta t}{1 - \frac{1}{2} r\Delta t} > 1$.

Of course, this is not a problem, since we have already found the analytical solution of the system.

Because of the rV-term in (24), together with the fact that $r > 0$, it might be tempting to conclude that no numerical scheme will be stable. This is not necessarily true. Take for example the Black-Scholes equation, where a rV-term also occurs. For this PDE, there are stable numerical schemes available. The interested reader is invited to consult [43].

4.4 Part 1: only the carbon price c is constant

A discretization of the domain on which we want to solve the PDE will lead to the grid illustrated in Figure 21.

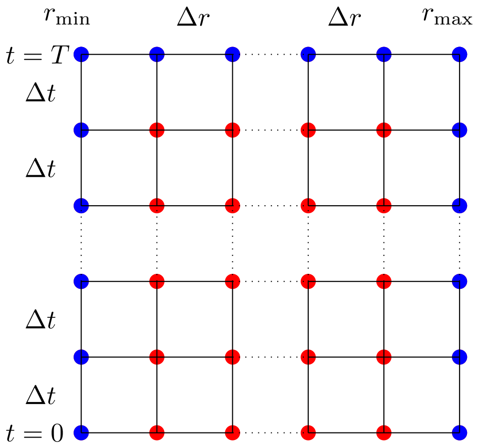


Figure 21: Discrization for the case where c is considered constant.

For this choice, we have that

$$\frac{\partial V}{\partial c} = \frac{\partial^2 V}{\partial c^2} = \frac{\partial^2 V}{\partial r \partial c} = 0.$$

This results in the following PDE:

$$\frac{\partial V}{\partial t}+\left(\alpha\left(\beta-r\right)-\lambda_r\sigma_r\sqrt{r}\right)\frac{\partial V}{\partial r}+\frac{1}{2}\sigma_r^2 r\frac{\partial^2 V}{\partial r^2}-rV=0. \tag{25}$$

The boundary conditions and the final condition are given by

- $V(t,r_{\min})=\left(c_0-Ke^{-r_{\min}(T-t)}\right)^+$,
- $V(t,r_{\max})=\left(c_0-Ke^{-r_{\max}(T-t)}\right)^+$,
- $V(T,r)=(c_0-K)^+$.

For this part of the analysis, we use the notation V_j^n for the nodes on the grid, where

- n is the time indicator: $n\xrightarrow{\Delta t=k}n+1$,
- j is the interest rate indicator: $j\xrightarrow{\Delta r=h}j+1$.

The next few (sub)sections are as follows: first, we will use forward differences for the first-order partial derivatives of (25). Then, a Von Neumann analysis is performed to obtain an expression for ξ. Although we will solve this PDE for multiple values of r, in the derivation of the numerical scheme, we will also consider r constant. This is necessary to perform a Von Neumann analysis since Von Neumann only works for linear PDEs.

After this, we will do the same for the case where we have backward differences for the first-order partial derivatives. From there, we will insert values for the parameters that occur in the expression for ξ and list the approximations that will be used before we conclude whether or not the numerical schemes are stable.

4.4.1 Part 1(a): constant carbon price, forward differences: derivation of the scheme

Here we use the following finite differences:

$$\begin{aligned}
\frac{\partial V}{\partial t}&:\ \text{forward difference: }\Delta_n V_j^n=V_j^{n+1}-V_j^n,\\
\frac{\partial V}{\partial r}&:\ \text{forward difference: }\Delta_j V_j^n=V_{j+1}^n-V_j^n,\\
\frac{\partial^2 V}{\partial r^2}&:\ \text{central difference: }\delta^2 V_j^n=V_{j-1}^n-2V_j^n+V_{j+1}^n.
\end{aligned}$$

If we substitute these three expressions into (25), and write

$$\gamma=\alpha\left(\beta-r\right)-\lambda_r\sigma_r\sqrt{r},$$

we are left with

$$\frac{V_j^{n+1}-V_j^n}{k}+\gamma\frac{V_{j+1}^n-V_j^n}{h}+\frac{1}{2}\sigma_r^2 r\frac{V_{j-1}^n-2V_j^n+V_{j+1}^n}{h^2}-rV_j^n=0.$$

We will bring all the V^n terms to the right-hand side and multiply with k.

$$V_j^{n+1} = V_j^n - \gamma k \frac{V_{j+1}^n - V_j^n}{h} - \frac{1}{2}\sigma_r^2 rk \frac{V_{j-1}^n - 2V_j^n + V_{j+1}^n}{h^2} + rkV_j^n.$$

Now, the V_j^{n+1} is isolated on the left-hand side. Then we collect all the V-terms, to obtain the following numerical scheme.

$$V_j^{n+1} = -\frac{\sigma_r^2 rk}{2h^2} V_{j-1}^n + \left(1 + \frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{h^2} + rk\right) V_j^n - \left(\frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{2h^2}\right) V_{j+1}^n. \tag{26}$$

4.4.2 Part 1(a): Von Neumann analysis

To perform a Von Neumann analysis, we write

$$V_j^n = \xi^n e^{\mathbf{i}\beta jh},$$

and we will substitute this expression into (26), which leads to the following expression:

$$\xi^{n+1} e^{\mathbf{i}\beta jh} = -\frac{\sigma_r^2 rk}{2h^2} \xi^n e^{\mathbf{i}\beta(j-1)h} + \left(1 + \frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{h^2} + rk\right) \xi^n e^{\mathbf{i}\beta jh} - \left(\frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{2h^2}\right) \xi^n e^{\mathbf{i}\beta(j+1)h}.$$

We multiply both sides of the above expression with $\xi^{-n} e^{-\mathbf{i}\beta jh}$ to obtain

$$\xi = -\frac{\sigma_r^2 rk}{2h^2} e^{-\mathbf{i}\beta h} + 1 + \frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{h^2} + rk - \left(\frac{\gamma k}{h} + \frac{\sigma_r^2 rk}{2h^2}\right) e^{\mathbf{i}\beta h}.$$

We rearrange the terms:

$$\xi = 1 + \frac{\gamma k}{h} + rk - \frac{\gamma k}{h} e^{\mathbf{i}\beta h} - \frac{\sigma_r^2 rk}{2h^2} \left(e^{-\mathbf{i}\beta h} - 2 + e^{\mathbf{i}\beta h}\right).$$

Using (22), we can simplify ξ a bit further:

$$\xi = 1 + \frac{\gamma k}{h} + rk - \frac{\gamma k}{h} e^{\mathbf{i}\beta h} + \frac{2\sigma_r^2 rk}{h^2} \sin^2\left(\frac{1}{2}\beta h\right).$$

4.4.3 Part 1(b): constant carbon price, backward differences: derivation of the scheme

Here we use the following finite differences:

$$\begin{aligned}
&\frac{\partial V}{\partial t} : \text{ backward difference: } \nabla_n V_j^n = V_j^n - V_j^{n-1},\\
&\frac{\partial V}{\partial r} : \text{ backward difference: } \nabla_j V_j^n = V_j^n - V_{j-1}^n,\\
&\frac{\partial^2 V}{\partial r^2} : \text{ central difference: } \delta^2 V_j^n = V_{j-1}^n - 2V_j^n + V_{j+1}^n.
\end{aligned}$$

If we substitute these three expressions into (25), and again write

$$\gamma = \alpha\left(\beta - r\right) - \lambda_r \sigma_r \sqrt{r},$$

we are left with

$$\frac{V_j^n - V_j^{n-1}}{k} + \gamma\frac{V_j^n - V_{j-1}^n}{h} + \frac{1}{2}\sigma_r^2 r\frac{V_{j-1}^n - 2V_j^n + V_{j+1}^n}{h^2} - rV_j^n = 0.$$

We increase the time index with 1 because these stencils are often given to go from the current state, state n to the next state, state $n+1$.

$$\frac{V_j^{n+1} - V_j^n}{k} + \gamma\frac{V_j^{n+1} - V_{j-1}^{n+1}}{h} + \frac{1}{2}\sigma_r^2 r\frac{V_{j-1}^{n+1} - 2V_j^{n+1} + V_{j+1}^{n+1}}{h^2} - rV_j^{n+1} = 0.$$

Multiplying both sides with k and bringing V_j^n to the right-hand side leads to the following expression:

$$V_j^{n+1} + \gamma k\frac{V_j^{n+1} - V_{j-1}^{n+1}}{h} + \frac{1}{2}\sigma_r^2 rk\frac{V_{j-1}^{n+1} - 2V_j^{n+1} + V_{j+1}^{n+1}}{h^2} - rkV_j^{n+1} = V_j^n.$$

For the next step, we collect all the V-terms, so we obtain a numerical scheme:

$$\left(\frac{\sigma_r^2 rk}{2h^2} - \frac{\gamma k}{h}\right)V_{j-1}^{n+1} + \left(1 + \frac{\gamma k}{h} - \frac{\sigma_r^2 rk}{h^2} - rk\right)V_j^{n+1} + \frac{\sigma_r^2 rk}{2h^2}V_{j+1}^{n+1} = V_j^n. \tag{27}$$

4.4.4 Part 1(b): Von Neumann analysis

To perform a Von Neumann analysis, we write

$$V_j^n = \xi^n e^{\mathbf{i}\beta jh},$$

and we will substitute this expression into (27), which leads to the following equality:

$$\left(\frac{\sigma_r^2 rk}{2h^2} - \frac{\gamma k}{h}\right)\xi^{n+1}e^{\mathbf{i}\beta(j-1)h} + \left(1 + \frac{\gamma k}{h} - \frac{\sigma_r^2 rk}{h^2} - rk\right)\xi^{n+1}e^{\mathbf{i}\beta jh} + \frac{\sigma_r^2 rk}{h^2}\xi^{n+1}e^{\mathbf{i}\beta(j+1)h} = \xi^n e^{\mathbf{i}\beta jh}.$$

We multiply both sides of this expression with $\xi^{-n}e^{-\mathbf{i}\beta jh}$:

$$\left(\frac{\sigma_r^2 rk}{2h^2} - \frac{\gamma k}{h}\right)\xi e^{-\mathbf{i}\beta h} + \left(1 + \frac{\gamma k}{h} - \frac{\sigma_r^2 rk}{h^2} - rk\right)\xi + \frac{\sigma_r^2 rk}{h^2}\xi e^{\mathbf{i}\beta h} = 1.$$

All three terms on the left-hand side contain a ξ, we use this and (22) to rewrite the left-hand side:

$$\left[1 + \frac{\gamma k}{h} - rk + \frac{\sigma_r^2 rk}{2h^2}\left(e^{-\mathbf{i}\beta h} - 2 + e^{\mathbf{i}\beta h}\right) - \frac{\gamma k}{h}e^{-\mathbf{i}\beta h}\right]\xi = 1,$$
$$\left[1 + \frac{\gamma k}{h} - rk - 2\frac{\sigma_r^2 rk}{h^2}\sin^2\left(\frac{1}{2}\beta h\right) - \frac{\gamma k}{h}e^{-\mathbf{i}\beta h}\right]\xi = 1.$$

4.4.5 Part 1: Insert values for the parameters

Although we assume r as a variable, we take a constant value for r, denoted by r_0, for this part of the Von Neumann analysis. This value is used to determine the coefficient matrix to perform a step backward in time. The other values that we will use to complete the analysis for both parts 1(a) and 1(b) follow the parameter calibration performed in from [29]:

- $\alpha = 0.91$,
- $\beta = 0.0451$,
- $r = r_0 = 0.05$,
- $\lambda_r = 0.01$, and
- $\sigma_r = 0.179$.

From here it follows that

$$\begin{aligned}\gamma &= \alpha\left(\beta - r_0\right) - \lambda_r \sigma_r \sqrt{r_0} \\ &= 0.91 \times (0.0451 - 0.05) - 0.01 \times 0.179 \times \sqrt{0.05} \\ &\approx -0.00486.\end{aligned}$$

Moreover, we assume that $r_{\min} = 0$ and $r_{\max} = 0.20$, a valid assumption for h would be

- $h = \frac{1}{100}\left(r_{\max} - r_{\min}\right) = 0.002$.

Of course, this conclusion about stability in the next sections heavily depends on the values of these parameters, i.e., the conclusion could change if we modify the value of some of the parameters. For now, we will only focus on the values that are given above.

In the upcoming analysis, we will use the following approximations that hold for 'small' values of x:

- $\sin(x) \approx x$,
- $\sin^2(x) \approx x^2$, and
- $\cos(x) \approx 1 - \frac{1}{2}x^2$.

4.4.6 Part 1(a): constant carbon price, forward differences: determination of the stability

As a reminder, we obtained the following expression for ξ, but we will replace r with r_0:

$$\begin{aligned}\xi &= 1 + \frac{\gamma k}{h} + r_0 k - \frac{\gamma k}{h} e^{\mathbf{i}\beta h} + \frac{2\sigma_r^2 r_0 k}{h^2} \sin^2\left(\frac{1}{2}\beta h\right) \\ &= 1 + \frac{\gamma k}{h} + r_0 k - \frac{\gamma k}{h}\left(\cos\left(\beta h\right) + \mathbf{i}\sin\left(\beta h\right)\right) + \frac{2\sigma_r^2 r_0 k}{h^2} \sin^2\left(\frac{1}{2}\beta h\right).\end{aligned}$$

Using the above-described approximations for the sine and cosine terms, we can write

$$\begin{aligned}\xi &\approx 1 + \frac{\gamma k}{h} + r_0 k - \frac{\gamma k}{h}\left(1 - \frac{1}{2}\beta^2 h^2\right) - \mathbf{i}\frac{\gamma k \beta h}{h} + \frac{2\sigma_r^2 r_0 k}{h^2}\frac{1}{4}\beta^2 h^2 \\ &= 1 + r_0 k + \frac{1}{2}\gamma k \beta^2 h - \mathbf{i}\gamma k \beta + \frac{1}{2}\sigma_r^2 r_0 \beta^2 k.\end{aligned}$$

For the chosen values of r_0, γ, h, and σ_r, we have that $r_0 k$ is the dominating term in the expression for ξ:

$$\xi \approx 1 + r_0 k. \tag{28}$$

Since $r_0 = 0.05 > 0$ this means that

$$\forall k > 0 : |\xi| \approx |1 + r_0 k| > 1,$$

which implies no stability for all positive time-step sizes k.

4.4.7 Part 1(b): constant carbon price, backward differences: determination of the stability

In the following analysis, we will again write r_0 instead of r. As a reminder, we obtained the following expression for ξ, which we will directly rewrite into a form with sine and cosine terms:

$$\left[1 + \frac{\gamma k}{h} - r_0 k - 2\frac{\sigma_r^2 r_0 k}{h^2} \sin^2\left(\frac{1}{2}\beta h\right) - \frac{\gamma k}{h} e^{-\mathbf{i}\beta h}\right] \xi = 1$$

$$\left[1 + \frac{\gamma k}{h} - r_0 k - 2\frac{\sigma_r^2 r_0 k}{h^2} \sin^2\left(\frac{1}{2}\beta h\right) - \frac{\gamma k}{h} \left(\cos\left(-\beta h\right) + \mathbf{i} \sin\left(-\beta h\right)\right)\right] \xi = 1$$

$$\left[1 + \frac{\gamma k}{h} - r_0 k - 2\frac{\sigma_r^2 r_0 k}{h^2} \sin^2\left(\frac{1}{2}\beta h\right) - \frac{\gamma k}{h} \left(\cos\left(\beta h\right) - \mathbf{i} \sin\left(\beta h\right)\right)\right] \xi = 1.$$

We make use of the approximation for the sine and cosine terms.

$$\left[1 + \frac{\gamma k}{h} - r_0 k - \frac{2\sigma_r^2 r_0 k}{h^2} \frac{1}{4}\beta^2 h^2 - \frac{\gamma k}{h} \left(1 - \mathbf{i}\beta h\right)\right] \xi = 1$$

$$\left[1 - r_0 k - \frac{1}{2}\sigma_r^2 \beta^2 r_0 k + \mathbf{i}\beta\gamma k\right] \xi = 1.$$

For the chosen values of r_0, γ and σ_r^2, we have that $-r_0 k$ is the dominating term in the above expression. Therefore, we can find an approximation for ξ.

$$\left[1 - r_0 k\right] \xi \approx 1. \tag{29}$$

For the absolute values of ξ, this means that

$$\forall k > 0 : |\xi| = \left|\frac{1}{1 - r_0 k}\right| > 1,$$

which implies that no matter the values of k, we don't have stability.

4.5 Part 2: only the interest rate r is constant

A discretization of the domain on which we want to solve the PDE will lead to the grid illustrated in Figure 22.

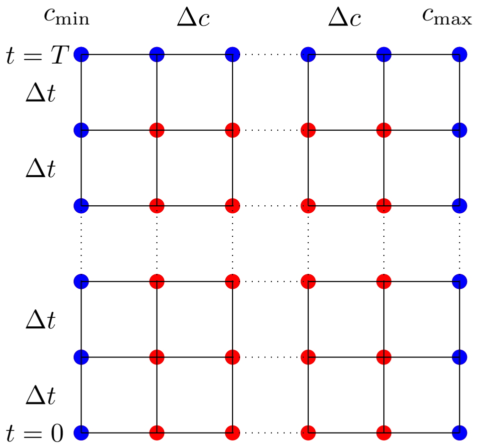


Figure 22: Discrization for the case where r is considered constant.

For this choice, we have that

$$\frac{\partial V}{\partial r} = \frac{\partial^2 V}{\partial r^2} = \frac{\partial^2 V}{\partial r \partial c} = 0.$$

This results in the following PDE:

$$\frac{\partial V}{\partial t} + (\mu c - \lambda_c \sigma_c c) \frac{\partial V}{\partial c} + \frac{1}{2} \sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} - rV = 0. \tag{30}$$

The boundary conditions and the final condition are given by

- $V(t, c_{\min}) = 0$,
- $V(t, c_{\max}) = \left(c_{\max} - Ke^{-r_0(T-t)}\right)^+$,
- $V(T, c) = (c_T - K)^+$.

For this part of the analysis, we use the notation V_m^n for the nodes on the grid, where

- n is the time indicator: $n \xrightarrow{\Delta t = k} n + 1$,
- m is the carbon price indicator: $m \xrightarrow{\Delta c = s} m + 1$.

The next few (sub)sections have the same structure as for Part 1: first, we will use forward differences for the first-order partial derivatives of (30). Then, a Von Neumann analysis is performed to obtain an expression for ξ. Although we will solve this PDE for multiple values of c, in the derivation of the numerical scheme, we will also consider c constant. This assumption is necessary for performing a Von Neumann analysis.

After that, we will do the same for the case where we have backward differences for the first-order partial derivatives. From there, we will insert values for the parameters that occur in the expression for ξ and list the approximations that will be used before we conclude whether the numerical schemes are stable.

4.5.1 Part 2(a): constant interest rate, forward differences: derivation of the scheme

Here we use the following finite differences:

$$\begin{aligned}
\frac{\partial V}{\partial t} &: \text{ forward difference: } \Delta_n V_m^n = V_m^{n+1} - V_m^n,\\
\frac{\partial V}{\partial c} &: \text{ forward difference: } \Delta_m V_m^n = V_{m+1}^n - V_m^n,\\
\frac{\partial^2 V}{\partial c^2} &: \text{ central difference: } \delta^2 V_m^n = V_{m-1}^n - 2V_m^n + V_{m+1}^n.
\end{aligned}$$

If we substitute these three expressions into (30), and write

$$\widetilde{\gamma} = \mu c - \lambda_c \sigma_c c,$$

then we are left with

$$\frac{V_m^{n+1} - V_m^n}{k} + \widetilde{\gamma}\frac{V_{m+1}^n - V_m^n}{s} + \frac{1}{2}\sigma_c^2 c^2 \frac{V_{m-1}^n - 2V_m^n + V_{m+1}^n}{s^2} - rV_m^n = 0.$$

We will multiply both sides of the above equality with k and bring some terms to the right-hand side:

$$V_m^{n+1} - V_m^n = -\widetilde{\gamma}k\frac{V_{m+1}^n - V_m^n}{s} - \frac{1}{2}\sigma_c^2 c^2 k \frac{V_{m-1}^n - 2V_m^n + V_{m+1}^n}{s^2} + rkV_m^n.$$

Next, we will bring V_m^n to the right-hand side. Then we collect all the V-terms, so that we obtain the following numerical scheme:

$$V_m^{n+1} = -\frac{\sigma_c^2 c^2 k}{2s^2} V_{m-1}^n + \left(1 + \frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2 c^2 k}{s^2} + rk\right) V_m^n - \left(\frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2 c^2 k}{2s^2}\right) V_{m+1}^n. \qquad (31)$$

4.5.2 Part 2(a): Von Neumann analysis

To perform a Von Neumann analysis, we write

$$V_m^n = \xi^n e^{\mathbf{i}\beta m s},$$

and we will substitute this expression into (31), which leads to the following expression:

$$\xi^{n+1} e^{\mathbf{i}\beta m s} = -\frac{\sigma_c^2 c^2 k}{2s^2}\xi^n e^{\mathbf{i}\beta(m-1)s} + \left(1 + \frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2 c^2 k}{s^2} + rk\right)\xi^n e^{\mathbf{i}\beta m s} - \left(\frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2 c^2 k}{2s^2}\right)\xi^n e^{\mathbf{i}\beta(m+1)s}.$$

We multiply both sides of the above expression with $\xi^{-n}e^{-\mathbf{i}\beta ms}$ to obtain

$$\begin{aligned}\xi &= -\frac{\sigma_c^2c^2k}{2s^2}e^{-\mathbf{i}\beta s} + 1 + \frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2c^2k}{s^2} + rk - \left(\frac{\widetilde{\gamma}k}{s} + \frac{\sigma_c^2c^2k}{2s^2}\right)e^{\mathbf{i}\beta s}.\\ &= -\frac{\sigma_c^2c^2k}{s^2}\left(e^{-\mathbf{i}\beta s} - 2 + e^{\mathbf{i}\beta s}\right) + 1 + rk + \frac{\widetilde{\gamma}k}{s}\left(1 - e^{\mathbf{i}\beta s}\right)\\ &= 2\frac{\sigma_c^2c^2k}{s^2}\sin^2\left(\frac{1}{2}\beta s\right) + 1 + rk + \frac{\widetilde{\gamma}k}{s}\left(1 - e^{\mathbf{i}\beta s}\right).\end{aligned}$$

In the final equality above, we used (22).

4.5.3 Part 2(b): constant interest rate, backward differences: derivation of the scheme

Here we use the following finite differences:

$$\begin{aligned}\frac{\partial V}{\partial t} &: \text{ backward difference: } \nabla_n V_m^n = V_m^n - V_j^{n-1},\\ \frac{\partial V}{\partial c} &: \text{ backward difference: } \nabla_m V_m^n = V_m^n - V_{m-1}^n,\\ \frac{\partial^2 V}{\partial c^2} &: \text{ central difference: } \delta^2 V_m^n = V_{m-1}^n - 2V_m^n + V_{m+1}^n.\end{aligned}$$

If we substitute these three expressions into (30), and again write

$$\widetilde{\gamma} = \mu c - \lambda_c\sigma_c c,$$

we are left with

$$\frac{V_m^n - V_m^{n-1}}{k} + \widetilde{\gamma}\frac{V_m^n - V_{m-1}^n}{s} + \frac{1}{2}\sigma_c^2c^2\frac{V_{m-1}^n - 2V_m^n + V_{m+1}^n}{s^2} - rV_m^n = 0.$$

We increase the time index with 1.

$$\frac{V_m^{n+1} - V_m^n}{k} + \widetilde{\gamma}\frac{V_m^{n+1} - V_{m-1}^{n+1}}{s} + \frac{1}{2}\sigma_c^2c^2\frac{V_{m-1}^{n+1} - 2V_m^{n+1} + V_{m+1}^{n+1}}{s^2} - rV_m^{n+1} = 0.$$

Multiplying both sides with k and bringing V_m^n to the right-hand side leads to the following expression:

$$V_m^{n+1} + \widetilde{\gamma}k\frac{V_m^{n+1} - V_{m-1}^{n+1}}{s} + \frac{1}{2}\sigma_c^2c^2k\frac{V_{m-1}^{n+1} - 2V_m^{n+1} + V_{m+1}^{n+1}}{s^2} - rkV_m^{n+1} = V_m^n.$$

For the next step, we collect all the V-terms to obtain a numerical scheme:

$$\left(\frac{\sigma_c^2c^2k}{2s^2} - \frac{\widetilde{\gamma}k}{s}\right)V_{m-1}^{n+1} + \left(1 + \frac{\widetilde{\gamma}k}{s} - \frac{\sigma_c^2c^2k}{s^2} - rk\right)V_m^{n+1} + \frac{\sigma_c^2c^2k}{2s^2}V_{m+1}^{n+1} = V_m^n. \tag{32}$$

4.5.4 Part 2(b): Von Neumann analysis

To perform a Von Neumann analysis, we write

$$V_m^n = \xi^n e^{\mathbf{i}\beta m s},$$

and we will substitute this expression into (32), which leads to the following expression:

$$\begin{aligned}\left(\frac{\sigma_c^2 c^2 k}{2s^2} - \frac{\widetilde{\gamma} k}{s}\right) \xi^{n+1} e^{\mathbf{i}\beta(m-1)s} + \left(1 + \frac{\widetilde{\gamma} k}{s} - \frac{\sigma_c^2 c^2 k}{s^2} - rk\right) \xi^{n+1} e^{\mathbf{i}\beta m s} \\ + \frac{\sigma_c^2 c^2 k}{2s^2} \xi^{n+1} e^{\mathbf{i}\beta(m+1)s} = \xi^n e^{\mathbf{i}\beta m s}.\end{aligned}$$

We multiply both sides of the above expression with $\xi^{-n} e^{-\mathbf{i}\beta m s}$ to obtain

$$\left(\frac{\sigma_c^2 c^2 k}{2s^2} - \frac{\widetilde{\gamma} k}{s}\right) \xi e^{-\mathbf{i}\beta s} + \left(1 + \frac{\widetilde{\gamma} k}{s} - \frac{\sigma_c^2 c^2 k}{s^2} - rk\right) \xi + \frac{\sigma_c^2 c^2 k}{2s^2} \xi e^{\mathbf{i}\beta s} = 1.$$

All three terms on the left-hand side contain a ξ, we use this together with (22) to rewrite the left-hand side:

$$\begin{aligned}\left[1 + \frac{\widetilde{\gamma} k}{s} - rk + \frac{\sigma_c^2 c^2 k}{2s^2} \left(e^{-\mathbf{i}\beta s} - 2 + e^{\mathbf{i}\beta s}\right) - \frac{\widetilde{\gamma} k}{s} e^{-\mathbf{i}\beta s}\right] \xi = 1, \\ \left[1 + \frac{\widetilde{\gamma} k}{s} - rk - 2\frac{\sigma_c^2 c^2 k}{s^2} \sin^2\left(\frac{1}{2}\beta s\right) - \frac{\widetilde{\gamma} k}{s} e^{-\mathbf{i}\beta s}\right] \xi = 1.\end{aligned}$$

4.5.5 Part 2: Insert values for the parameters

In [29], a table of values for some parameters is provided. We can use this table to continue the analysis for both parts 2(a) and 2(b).

- $\mu = 0.058$,
- $r = r_0 = 0.05$,
- $\lambda_c = 0.2$, and
- $\sigma_c = 0.832$.

Although we assume c as a variable, we take a constant value for c, denoted by c_0, for this part of the Von Neumann analysis. This value is used to determine the coefficient matrix to perform a step backward in time. We assume that $c_{\min} = 20$ and $c_{\max} = 120$, we assume that

- $c_0 = \frac{1}{2}\left(c_{\max} + c_{\min}\right) = \frac{1}{2}(120 + 20) = 70$, and
- $s = \frac{1}{100}\left(c_{\max} - c_{\min}\right) = \frac{1}{100}(120 - 20) = 1$.

From here it follows that

$$\begin{aligned}\widetilde{\gamma} &= \mu c_0 - \lambda_c \sigma_c c_0 \\ &= 0.058 \times 70 - 0.2 \times 0.832 \times 70 \\ &\approx -7.588.\end{aligned}$$

For Part 1, we already stated that the conclusion about stability in the next sections heavily depends on the values of these parameters above. For now, we will only focus on the values above, but keep in mind that the stability can get better or worse if some of the parameter values are modified.

We again use the Taylor series for the sine and cosine terms in the upcoming analysis, i.e.

- $\sin(x) \approx x$,
- $\sin^2(x) \approx x^2$, and
- $\cos(x) \approx 1 - \frac{1}{2}x^2$.

Note that since $s = 1 \gg 0$, these approximations are not as precise as for Part 1, but it is still enough to draw a valid conclusion about the stability.

4.5.6 Part 2(a): constant interest rate, forward differences: determination of the stability

We follow the same steps as for Part 1, which means in particular that we will write c_0 instead of c. As a reminder, we have the following scheme when we use forward differences:

$$\begin{aligned}\xi &= 2\frac{\sigma_c^2 c^2 k}{s^2}\sin^2\left(\frac{1}{2}\beta s\right) + 1 + r_0 k + \frac{\widetilde{\gamma} k}{s}\left(1 - e^{\mathbf{i}\beta s}\right)\\ &= 2\frac{\sigma_c^2 c^2 k}{s^2}\sin^2\left(\frac{1}{2}\beta s\right) + 1 + r_0 k + \frac{\widetilde{\gamma} k}{s}\left(1 - \left(\cos\left(\beta s\right) + \mathbf{i}\sin\left(\beta s\right)\right)\right).\end{aligned}$$

For the next part of the analysis, again we use the Taylor series for the sine and cosine terms.

$$\begin{aligned}\xi &\approx \tilde{2}\frac{\sigma_c^2 c_0^2 k}{s^2}\left(\frac{1}{2}\beta s\right)^2 + 1 + rk + \frac{\widetilde{\gamma} k}{s}\left(1 - 1 + \frac{1}{2}\beta^2 s^2 - \mathbf{i}\beta s\right)\\ &= \frac{1}{2}\sigma_c^2 c_0^2 \beta^2 k + 1 + r_0 k + \frac{1}{2}\widetilde{\gamma} k \beta^2 s - \widetilde{\gamma} k \beta \mathbf{i}.\end{aligned}$$

If we look at the real part of this expression, denoted by $\Re\left(\xi\right)$, and keep the values of the parameter in mind, we make the following observation:

$$\Re\left(\xi\right) = \frac{1}{2}\sigma_c^2 c_0^2 \beta^2 k + 1 + r_0 k + \frac{1}{2}\widetilde{\gamma} k \beta^2 s = \underbrace{1 + r_0 k}_{>1} + \frac{1}{2}\beta^2 k \underbrace{\left(\sigma_c^2 c_0^2 + \widetilde{\gamma} s\right)}_{\gg 0} \tag{33}$$

This leads to the conclusion that $\Re\left(\xi\right) > 1$, and this implies that for all $k > 0$, we have that $|\xi| > 1$, which implies that we do not have stability.

4.5.7 Part 2(b): constant interest rate, backward differences: determination of the stability

$$\left[1 + \frac{\widetilde{\gamma} k}{s} - rk - 2\frac{\sigma_c^2 c_0^2 k}{s^2}\sin^2\left(\frac{1}{2}\beta s\right) - \frac{\widetilde{\gamma} k}{s}e^{-\mathbf{i}\beta s}\right]\xi = 1$$

$$\left[1 + \frac{\widetilde{\gamma} k}{s} - rk - 2\frac{\sigma_c^2 c_0^2 k}{s^2}\sin^2\left(\frac{1}{2}\beta s\right) - \frac{\widetilde{\gamma} k}{s}\left(\cos\left(\beta s\right) + \mathbf{i}\sin\left(-\beta s\right)\right)\right]\xi = 1.$$

Using the Taylor series for the sine and cosine terms, we obtain an expression that can be simplified.

$$\left[1+\frac{\widetilde{\gamma}k}{s}-rk-2\frac{\sigma_c^2c_0^2k}{s^2}\left(\frac{1}{2}\beta s\right)^2-\frac{\widetilde{\gamma}k}{s}\left(1-\frac{1}{2}\beta^2s^2-\mathbf{i}\beta s\right)\right]\xi=1$$
$$\left[1+\frac{\widetilde{\gamma}k}{s}-rk-2\frac{\sigma_c^2c_0^2k}{s^2}\left(\frac{1}{2}\beta s\right)^2-\frac{\widetilde{\gamma}k}{s}+\frac{1}{2}\widetilde{\gamma}k\beta^2s+\widetilde{\gamma}k\beta\mathbf{i}\right]\xi=1$$
$$\left[1-rk-\frac{1}{2}\sigma_c^2c_0^2\beta^2k+\frac{1}{2}\widetilde{\gamma}k\beta^2s+\widetilde{\gamma}k\beta\mathbf{i}\right]\xi=1.$$

If we insert the values, we find that the dominating term in the above expression is the $\frac{1}{2}\sigma_c^2c_0^2\beta^2k$-term. Therefore, we find that

$$|\xi|\approx\left|\frac{1}{-\frac{1}{2}\sigma_c^2c_0^2\beta^2k}\right|=\frac{1}{\frac{1}{2}\sigma_c^2c_0^2\beta^2k}. \tag{34}$$

We have stability if and only if

$$-1\leq\frac{1}{\frac{1}{2}\sigma_c^2c_0^2\beta^2k}\leq 1.$$

The left inequality is automatically satisfied since both the numerator and denominator are positive for all $k>0$. The right inequality gives us a lower bound for k:

$$k\geq\frac{1}{\frac{1}{2}\sigma_c^2c_0^2\beta^2}=\frac{2}{\sigma_c^2c_0^2\beta^2}=\frac{2}{0.832^2\cdot 70^2\cdot\pi^2}\approx 0.00006.$$

4.6 Part 3: non-constant r and c

To simplify the derivation of the numerical scheme, we assume that the correlation coefficient between the interest rate and the carbon price is equal to zero, i.e., $\rho=0$. Using the same notations again

$$\gamma=\alpha\left(\beta-r_0\right)-\lambda_r\sigma_r\sqrt{r_0},\text{ and}$$
$$\widetilde{\gamma}=\mu c_0-\lambda_c\sigma_c c_0,$$

then, the green bond PDE (24) is reduced to:

$$\frac{\partial V}{\partial t}+\gamma\frac{\partial V}{\partial r}+\widetilde{\gamma}\frac{\partial V}{\partial c}+\frac{1}{2}\sigma_c^2c_0^2\frac{\partial^2V}{\partial c^2}+\frac{1}{2}\sigma_r^2r_0\frac{\partial^2V}{\partial r^2}-r_0V=0. \tag{35}$$

The boundary conditions remain the same as for the case where $\rho\neq 0$. For the discretization, we will use backward differences and central differences. Since we have a final condition, we can compute the 'previous state' directly, i.e., without solving a system of equations. Therefore, this results in an explicit method. To give an overview of the finite differences:

$$\begin{aligned}
\frac{\partial V}{\partial t} &: \text{ backward difference: } \nabla_n V^n_{j,m} = V^n_{j,m} - V^{n-1}_{j,m},\\
\frac{\partial V}{\partial r} &: \text{ backward difference: } \nabla_j V^n_{j,m} = V^n_{j,m} - V^n_{j-1.m},\\
\frac{\partial V}{\partial c} &: \text{ backward difference: } \nabla_m V^n_{j,m} = V^n_{j,m} - V^n_{j,m-1},\\
\frac{\partial^2 V}{\partial c^2} &: \text{ central difference: } \delta^2_m V^n_{j,m} = V^n_{j,m-1} - 2V^n_{j,m} + V^n_{j,m+1},\\
\frac{\partial^2 V}{\partial r^2} &: \text{ central difference: } \delta^2_j V^n_{j,m} = V^n_{j-1,m} - 2V^n_{j,m} + V^n_{j+1,m}.
\end{aligned}$$

Now we can discretize (35), increasing the time index again with 1.

$$\begin{aligned}
\frac{V^{n+1}_{j,m} - V^n_{j,m}}{k} + \gamma \frac{V^{n+1}_{j,m} - V^{n+1}_{j-1.m}}{h} + \widetilde{\gamma} \frac{V^{n+1}_{j,m} - V^{n+1}_{j,m-1}}{s} + \frac{1}{2}\sigma_c^2 c_0^2 \frac{V^{n+1}_{j,m-1} - 2V^{n+1}_{j,m} + V^{n+1}_{j,m+1}}{s^2}\\
+ \frac{1}{2}\sigma_r^2 r_0 \frac{V^{n+1}_{j-1,m} - 2V^{n+1}_{j,m} + V^{n+1}_{j+1,m}}{h^2} - r_0 V^{n+1}_{j,m} = 0.
\end{aligned}$$

We multiply both sides with k. Moreover, we take $V^n_{j,m}$ to the right-hand side.

$$\begin{aligned}
V^{n+1}_{j,m} + \gamma k \frac{V^{n+1}_{j,m} - V^{n+1}_{j-1.m}}{h} + \widetilde{\gamma} k \frac{V^{n+1}_{j,m} - V^{n+1}_{j,m-1}}{s} + \frac{1}{2}\sigma_c^2 c_0^2 k \frac{V^{n+1}_{j,m-1} - 2V^{n+1}_{j,m} + V^{n+1}_{j,m+1}}{s^2}\\
+ \frac{1}{2}\sigma_r^2 r_0 k \frac{V^{n+1}_{j-1,m} - 2V^{n+1}_{j,m} + V^{n+1}_{j+1,m}}{h^2} - r_0 V^{n+1}_{j,m} = V^n_{j,m}.
\end{aligned}$$

Now we collect the terms to obtain the numerical scheme:

$$\begin{aligned}
\left(1 + \frac{\widetilde{\gamma} k}{s} + \frac{\gamma k}{h} - \frac{\sigma_c^2 c_0^2 k}{s^2} - \frac{\sigma_r^2 r_0 k}{h^2} - r_0 k\right) V^{n+1}_{j,m} + \left(\frac{\sigma_c^2 c_0^2 k}{2s^2} - \frac{\widetilde{\gamma} k}{s}\right) V^{n+1}_{j,m-1}\\
+ \left(\frac{\sigma_r^2 r_0 k}{2h^2} - \frac{\gamma k}{h}\right) V^{n+1}_{j-1,m} + \frac{\sigma_c^2 c_0^2 k}{2s^2} V^{n+1}_{j,m+1} + \frac{\sigma_r^2 r_0 k}{2h^2} V^{n+1}_{j+1,m} = V^n_{j,m}.
\end{aligned} \tag{36}$$

Performing a Von Neumann analysis will not be necessary. When we choose backward differences for the first-order partial derivatives, we have already seen that the numerical scheme for a constant value of c (Chapter 4.5.6) is not stable. Moving from a 2-dimensional to a 3-dimensional scheme will only have a negative contribution to the stability because of the extra terms.

5 Numerical Results

This chapter will provide numerical evidence for the claims in Chapter 4. The Python code for Parts 1(a,b), 2(a,b) and 3 can be found in Appendix B. The pieces of code for the implicit methods contain the iterative solution methods and a direct solution method, namely, the inverse matrix. Note that the numerical schemes for the implicit methods, (26) and (31), are not symmetric. Therefore, the corresponding matrix is not SPD, so the Conjugate Gradient method cannot be used to solve this system of linear equations. Only the Bi-CGSTAB and the GMRES method can be used. No matter the chosen solution method, all the figures in this chapter will look the same, i.e., will lead to approximately the same green bond values, also called coupon values. For this reason, we did not consider using a preconditioner matrix M for the iterative solution methods. The interested reader who wants to compare the Bi-CGSTAB and the GMRES method is invited to consult [29].

A lot of different parameters appear in the model. We use the analysis performed in [29] to find appropriate values for all these parameters. Unless specified otherwise, the values of the parameters are as given in Table 1.

Parameter	Value	Parameter	Value
α	0.91	c_0	70
β	0.0451	K	50
μ	0.058	λ_r	0.01
t_{min}	0	λ_c	0.2
$t_{\text{max}} = T$	1	σ_r	0.179
r_{min}	0	σ_c	0.832
r_{max}	0.20	$\Delta t = k$	0.001
r_0	0.05	$\Delta r = h$	0.01
c_{min}	20	$\Delta c = s$	1
c_{max}	120	ρ	0

Table 1: Values for the parameters in the green bond PDE (24).

5.1 Part 1: constant carbon price

As a reminder, for Part 1 we consider the carbon price c to be constant. Therefore, all the partial derivatives with respect to c vanish. For this assumption, the green bond PDE (24) can be written as

$$\frac{\partial V}{\partial t} + \left(\alpha\left(\beta - r_0\right) - \lambda_r \sigma_r \sqrt{r_0}\right)\frac{\partial V}{\partial r} + \frac{1}{2}\sigma_r^2 r_0 \frac{\partial^2 V}{\partial r^2} - r_0 V = 0.$$

Substituting the values of the parameters in the PDE leaves us with

$$\frac{\partial V}{\partial t} - 0.0049\frac{\partial V}{\partial r} + 0.0008\frac{\partial^2 V}{\partial r^2} - 0.05V = 0.$$

The boundary conditions and the final condition are given by

- $V(t, r_{\text{min}}) = \left(c_0 - Ke^{-r_{\text{min}}(T-t)}\right)^+ = (70 - 50)^+ = 20,$
- $V(t, r_{\text{max}}) = \left(c_0 - Ke^{-r_{\text{max}}(T-t)}\right)^+ = 70 - 50e^{-0.20(1-t)},$
- $V(T, r) = (c_0 - K)^+ = (70 - 50)^+ = 20.$

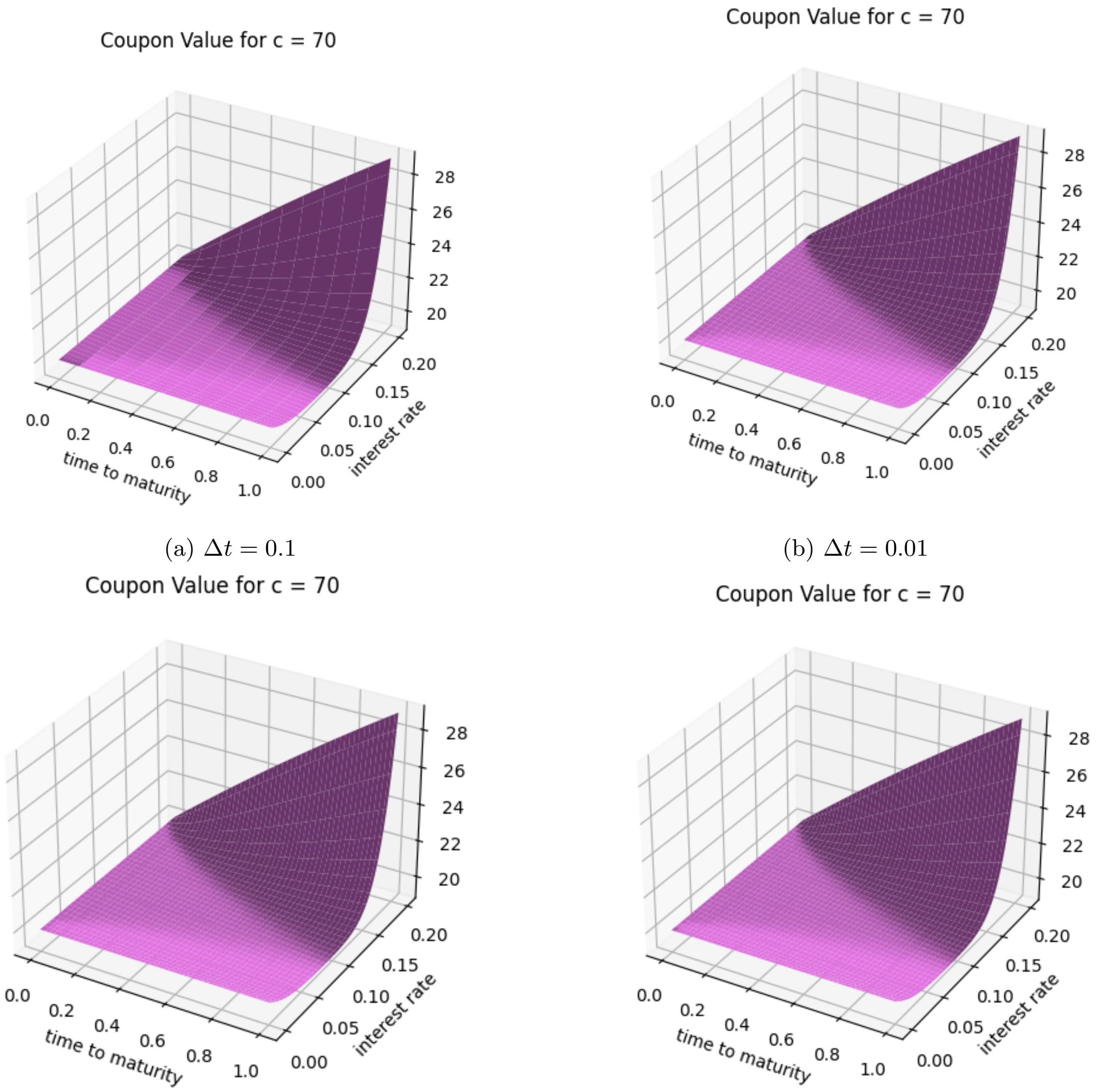
The Python code for both Part 1(a) and 1(b) provides a plot of the coupon value and a matrix with the coupon values. This matrix should be read as follows: the first row of values corresponds to the case where $t = T$. The second row corresponds to $t = T - \Delta t$, up till the last row, corresponding to $t = 0$. Similarly, the first column corresponds to the case where $r = r_{\text{min}}$. The second column corresponds to $r = r_{\text{min}} + \Delta r$, up till the last column, corresponding to the case where $r = r_{\text{max}}$.

5.1.1 Part 1(a): constant carbon price, forward differences

In Chapter 4.4.2, we performed a Von Neumann analysis and in Chapter 4.4.6, we concluded that the numerical scheme is not stable, since equation (28) gave us an approximation of the absolute value of ξ that was strictly greater than 1:

$$\forall k \geq 0 : |\xi| = |1 + r_0 k| > 1.$$

Thus: no matter the value of the time step size $\Delta t = k$, the scheme will not be (absolutely) stable. Although the numerical scheme is not stable, the plots of the coupon value are what we expect them to look like, i.e., the plots provide reasonable coupon values. This holds for different values of Δt, ranging from 0.1 to 0.0001, see Figure 23.



(a) $\Delta t = 0.1$

(b) $\Delta t = 0.01$

(c) $\Delta t = 0.001$

(d) $\Delta t = 0.0001$

Figure 23: Part 1(a): constant carbon price, forward differences:
Coupon values for different values of Δt.

One reason for this phenomenon might be the fact that although we have that $|\xi| > 1$, the amplification factor is not much larger than 1. Since $r_0 = 0.05$, the values for $|\xi|$ for multiple values of Δt are displayed in Table 2.

$\Delta t = k$	$\xi = 1 + r_0 k$
0.1	1.005
0.01	1.0005
0.001	1.00005
0.0001	1.000005

Table 2: Part 1(a): constant carbon price, forward differences: values for the amplification factor ξ for $r_0 = 0.05$

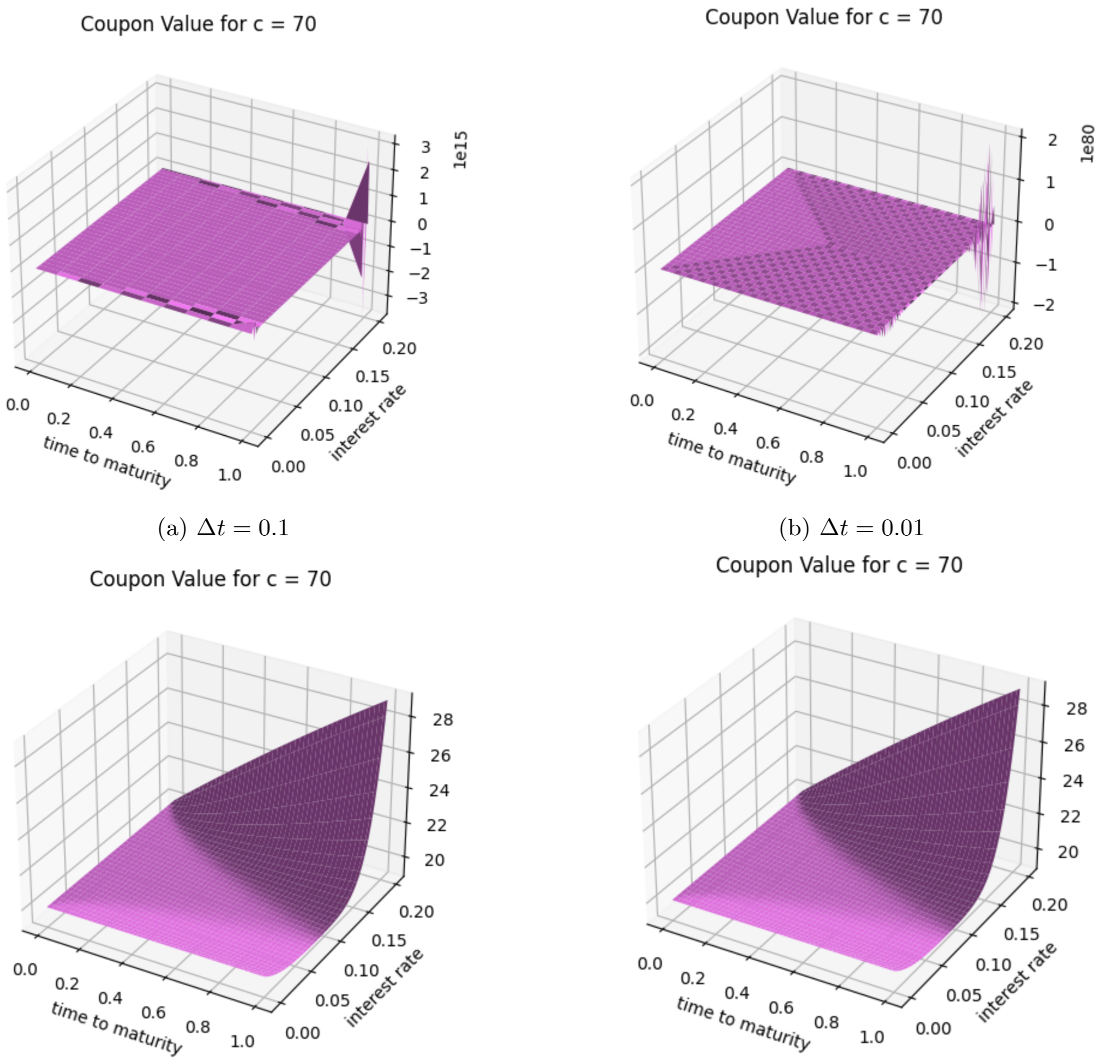
Another reason that the figures look so well could be the small coefficients of the partial derivatives (first- and second-order) with respect to r. Due to this, the error might not grow fast enough to observe real instability.

5.1.2 Part 1(b): constant carbon price, backward differences

In Chapter 4.4.7, we concluded that the numerical scheme is not stable:

$$\forall k > 0 : |\xi| = \left| \frac{1}{1 - r_0 k} \right| > 1.$$

Therefore, we must choose a small value of Δt, i.e., a large number of steps in the time direction, to obtain reasonable graphs. This is illustrated in Figure 24, where the plots become more and more accurate the smaller Δt gets.



(a) $\Delta t = 0.1$

(b) $\Delta t = 0.01$

(c) $\Delta t = 0.001$

(d) $\Delta t = 0.0001$

Figure 24: Part 1(b): constant carbon price, backward differences:
Coupon value for different values of Δt.

Unlike Part 1(a), choosing the time step size equal to 0.1 or 0.01 is not small enough to result in feasible coupon values. Only smaller values will work, see Figure 24. The explanation for the bad quality of Figures 24a and 24b cannot be found in the amplification factor ξ, since this is, like Part 1(a), close to 1 for $\Delta t \in \{0.1, 0.01, 0.001, 0.0001\}$, see Table 3 below.

$\Delta t = k$	$\xi = \frac{1}{1-r_0 k}$
0.1	1.005025
0.01	1.000500
0.001	1.000050
0.0001	1.000005

Table 3: Part 1(b): constant carbon price, backward differences: values for the amplification factor ξ for $r_0 = 0.05$

The unrealistic coupon values in Figure 24a and Figure 24b could be caused by the boundary conditions. The Von Neumann analysis does not consider the boundary conditions, so the effect of the boundary conditions on the stability is not known. It might be the case that the boundary conditions cause more instability to explicit methods, such as Part 1(b), than implicit methods, such as Part 1(a).

It is worthwhile to mention that we observe the same behavior for different values of the carbon price (c_0) and the strike price (K), as long as the ratio between these two parameters stays about the same. This is because c_0 and K occur together in the boundary conditions and the final condition.

5.2 Part 2: constant interest rate

As a reminder, for Part 2 we consider the interest rate r to be constant. Therefore, all the partial derivatives with respect to r vanish. For this assumption, the green bond PDE (24) can be written as

$$\frac{\partial V}{\partial t} + (\mu c_0 - \lambda_c \sigma_c c_0)\frac{\partial V}{\partial c} + \frac{1}{2}\sigma_c^2 c_0^2 \frac{\partial^2 V}{\partial c^2} - r_0 V = 0.$$

Substituting the values in the PDE leaves us with

$$\frac{\partial V}{\partial t} - 7.588\frac{\partial V}{\partial c} + 1695.9\frac{\partial^2 V}{\partial c^2} - 0.05V = 0.$$

The boundary conditions and the final condition are given by

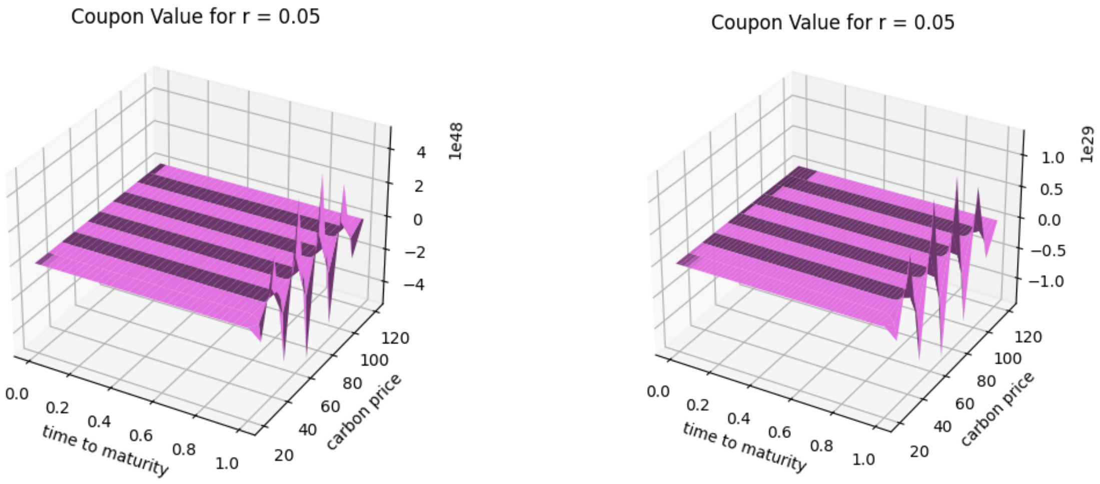
- $V(t, c_{\min}) = 0$,
- $V(t, c_{\max}) = \left(c_{\max} - Ke^{-r_0(T-t)}\right)^+ = 120 - 50e^{-0.05(1-t)}$,
- $V(T, c) = (c_T - K)^+ = (c_T - 50)^+$.

As for Part 1, the Python code for both Part 2(a) and 2(b) provides a plot of the coupon value and a matrix with the coupon values. This matrix should be read as follows: the first row of values corresponds to the case where $t = T$. The second row corresponds to $t = T - \Delta t$, up till the last row, corresponding to $t = 0$. Instead of the interest rate r, the first column for the discretization for Part 2 corresponds to the case where $c = c_{\min}$. The second column corresponds to $c = c_{\min} + \Delta c$, up till the last column, corresponding to the case where $c = c_{\max}$.

For both Part 2(a) and 2(b), we were forced to change the value of Δc from 1 to 10. A value of Δc smaller than 10 will lead to computational errors.

5.2.1 Part 2(a): constant interest rate, forward differences

As for Part 1(a) and 1(b), we conclude that the numerical scheme in Part 2(a) was not stable. The main difference that we observe here is that no matter how small we choose Δt, we do not obtain pictures that accurately describe the coupon value.



(a) $\Delta t = 0.01$

(b) $\Delta t = 0.0001$

Figure 25: Part 2(a): constant interest rate, forward differences:
Coupon value for different values of Δt, where $c_0 = 70$ and $K = 50$.

To find a reason for this behavior, we first look at the amplification factor ξ. As a reminder, in Chapter 4.5.6, we found that

$$\Re(\xi) = \underbrace{1 + r_0 k}_{>1} + \frac{1}{2}\beta^2 k \underbrace{\left(\sigma_c^2 c_0^2 + \widetilde{\gamma} s\right)}_{>0} \Longrightarrow \Re(\xi) > 1,$$

where the value of β was equal to π. This led to the conclusion that $|\xi| > 1$. Whereas the value of ξ was close to 1 for Part 1(a) and 1(b), now we have that

$$\begin{aligned}\Re(\xi) &= 1 + r_0 k + \frac{1}{2}\beta^2 k \left(\sigma_c^2 c_0^2 + \widetilde{\gamma} s\right) \\ &\approx \begin{cases} 164.34, \text{ if } k = 0.01, \\ 2.64, \text{ if } k = 0.0001. \end{cases}\end{aligned}$$

In both cases, we observe that the (real part of the) amplification factor is much bigger than 1. This explains the shape of the graphs in Figure 25.

Another explanation for the badly shaped graphs in Figure 25 can be found when we examine the PDE, which, as a reminder, is given by

$$\frac{\partial V}{\partial t} + (\mu c - \lambda_c \sigma_c c)\frac{\partial V}{\partial c} + \frac{1}{2}\sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} - rV = 0.$$

In Chapter 4.5.5, after writing down the first steps of the Von Neumann analysis, we inserted values for the parameters. This led to the following coefficients for the partial derivatives with respect to c:

- $\widetilde{\gamma} := \mu c - \lambda_c \sigma_c c \approx -7.588$, and
- $\eta := \frac{1}{2}\sigma_c^2 c^2 \approx 1695.9$.

It turns out that the approximations for the coupon values become more accurate if the ratio $|\widetilde{\gamma}/\eta|$ is smaller. We can do this by choosing a smaller value for the initial carbon price c_0. The boundary conditions must also be considered, and since c_0 and the strike price K occur together in these boundary conditions, K should also be modified. By choosing smaller values for c_0 and K, the amplification factor ξ takes values closer to 1. As a result, the stability of the numerical scheme increases. Recall that the formula for the amplification factor for this numerical scheme is given by

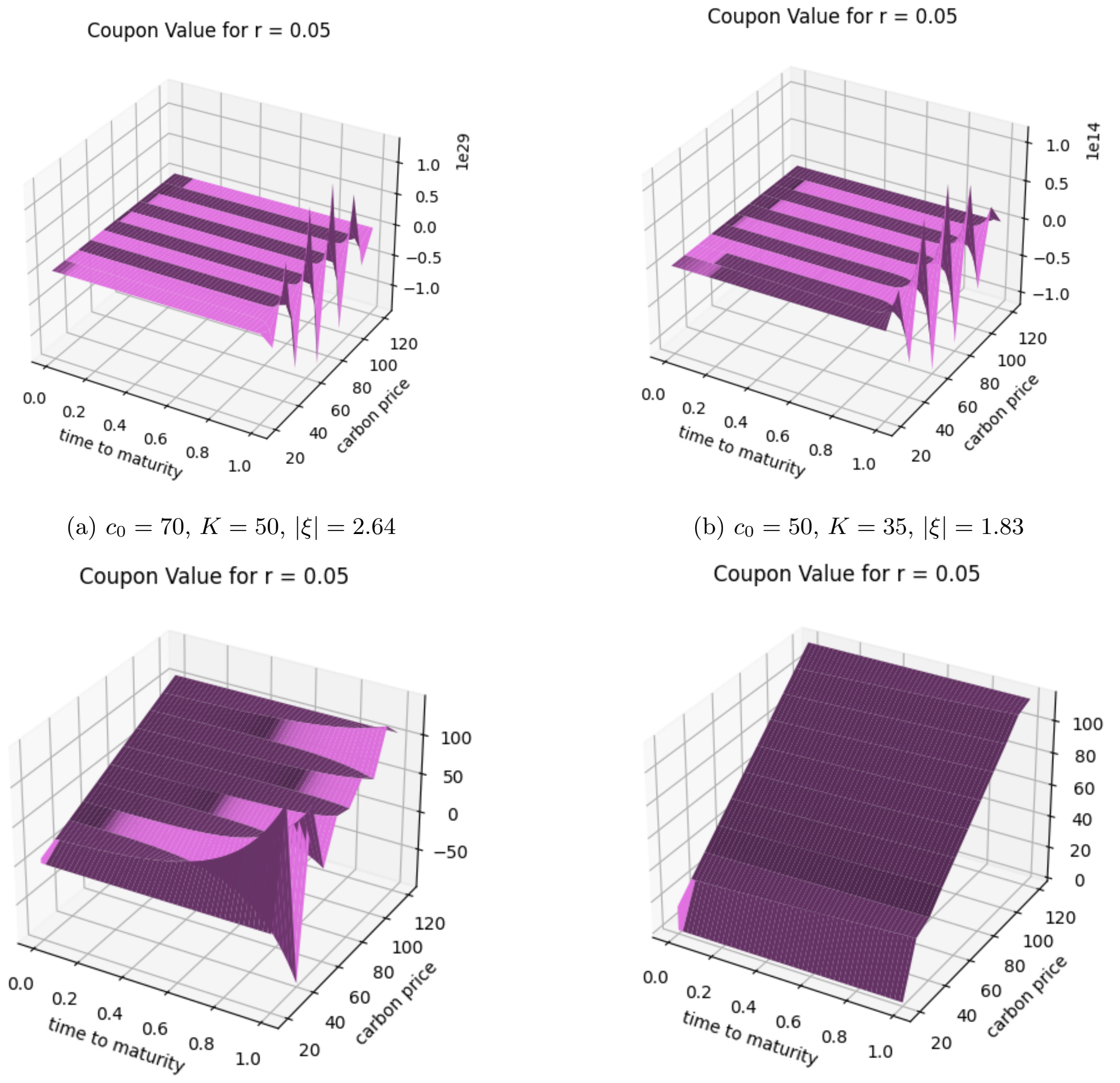
$$\xi = \frac{1}{2}\sigma_c^2 c_0^2 \beta^2 k + 1 + r_0 k + \frac{1}{2}\widetilde{\gamma} k \beta^2 s - \widetilde{\gamma} k \beta \mathbf{i}.$$

In Table 4, we have computed the ratio $|\widetilde{\gamma}/\eta|$ and the absolute value of the amplification factor $|\xi|$ for multiple values of c_0 and K, where the value of Δt is equal to 0.0001.

c_0	K	$\widetilde{\gamma}$	η	$\lvert\eta/\widetilde{\gamma}\rvert$	$\lvert\xi\rvert$
70	50	-7.588	1695.9	223.5	2.64
50	35	-5.42	865.28	159.6	1.83
20	14	-2.17	138.44	63.9	1.13
7	5	-0.76	16.96	22.4	1.01

Table 4: Coefficients for the green bond PDE, (30), where r is constant.

For each of the choices for c_0 and K, a plot of the coupon value is made and can be seen in Figure 26 below. For each plot, we have that $\Delta t = 0.0001$. Observe that the quality graphs of the coupon values increase when the amplification factor gets closer to 1.



(a) $c_0 = 70$, $K = 50$, $|\xi| = 2.64$

(b) $c_0 = 50$, $K = 35$, $|\xi| = 1.83$

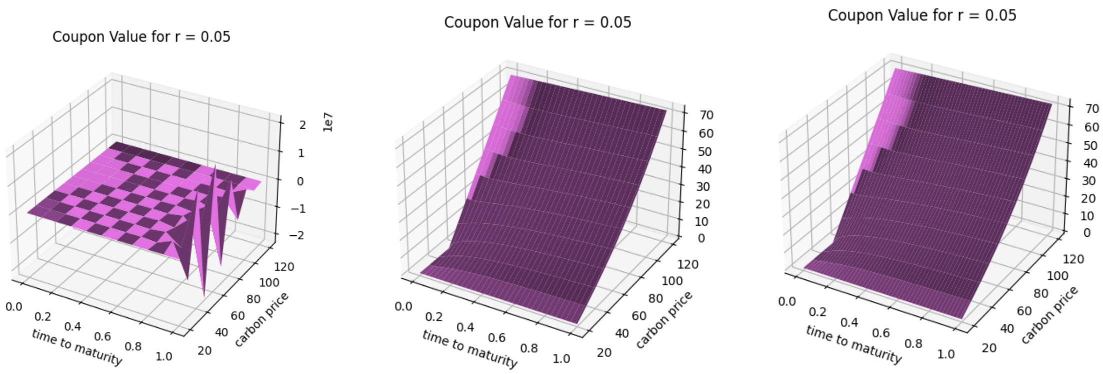
(c) $c_0 = 20$, $K = 14$, $|\xi| = 1.13$

(d) $c_0 = 7$, $K = 5$, $|\xi| = 1.01$

Figure 26: Part 2(a): constant interest rate, forward difference:
Coupon value for different initial values of c_0 and K.

5.2.2 Part 2(b): constant interest rate, backward differences

In Chapter 4.5.7, the Von Neumann analysis allowed us to conclude that the numerical scheme is stable if $\Delta t = k \geq 0.00006$. After numerical simulations, we observed that this conclusion is not completely valid. In Figure 27, we see that the graph of the coupon value is not nicely shaped for $\Delta t = 0.1$. For Δt values of 0.01 and 0.001, we don't have to modify the values of c_0 and K to obtain reasonable plots for the coupon values, unlike Part 2(a).



(a) $\Delta t = 0.1$ (b) $\Delta t = 0.01$ (c) $\Delta t = 0.001$

Figure 27: Part 2(b): constant interest rate, backward differences:
Coupon value for different values of Δt.

In Figure 27, we see that for $\Delta t = 0.1$, we do not end up with reasonable coupon values, although $0.1 \geq 0.00006$. This is not the only observation that does not correspond with the conclusion drawn after the Von Neumann analysis. In Figure 28, we choose a Δt value smaller than 0.00006, and by the Von Neumann analysis, we should not have stability. The Figure on the other hand is nicely shaped. This can caused by the change of Δc from 1 to 10. Therefore, the approximations used in the Von Neumann analysis in Chapter 4.5.7 become less accurate and it could very well be the case we do have stability.

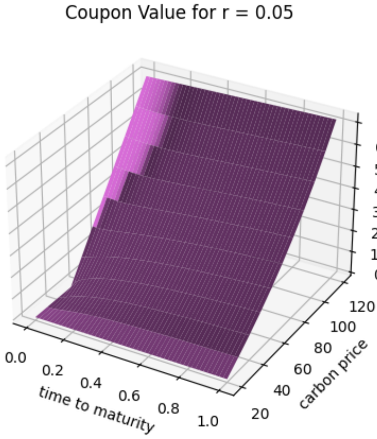


Figure 28: Part 2(b): constant interest rate, backward differences:
$\Delta t = 0.00001$

An explanation for the fact that the analytical results do not match the numerical results can be found in the computation of the amplification factor in Chapter 4.5.6. Here we used Taylor approximations for the sine and cosine terms. Normally, approximations for $\cos(x)$ and $\sin(x)$ are only accurate for small x. In our analysis, Δc occurred inside the sine and cosine terms. For the numerical part of this thesis, this parameter is modified from 1 to 10. Therefore, the approximations become less accurate.

5.3 Part 3: neither interest rate nor carbon price is constant

We assume that both r and c are not constant for this part of the analysis. Unlike Part 1 and 2, no partial derivatives vanish and we are left with a three-dimensional PDE, which, as a reminder, is given below.

$$\begin{aligned}\frac{\partial V}{\partial t} &+ (\mu c - \lambda_c \sigma_c c)\frac{\partial V}{\partial c} + \left(\alpha(\beta - r) - \lambda_r \sigma_r \sqrt{r}\right)\frac{\partial V}{\partial r} \\ &+ \frac{1}{2}\left(\sigma_c^2 c^2 \frac{\partial^2 V}{\partial c^2} + \sigma_r^2 r \frac{\partial^2 V}{\partial r^2} + 2c\rho\sigma_c\sigma_r\sqrt{r}\frac{\partial^2 V}{\partial r \partial c}\right) - rV = 0.\end{aligned}$$

In the parameter calibration performed in [29], it followed that $\rho = 0.2$. To make life a little bit more easy, we assumed that $\rho = 0$. Using this value for ρ, the cross partial derivative drops out of the PDE. Moreover, we have written

- $\gamma = \alpha\left(\beta - r_0\right) - \lambda_r \sigma_r \sqrt{r_0}$, and
- $\widetilde{\gamma} = \mu c_0 - \lambda_c \sigma_c c_0$.

Using these notations, we are left with a PDE given by

$$\frac{\partial V}{\partial t} + \gamma \frac{\partial V}{\partial r} + \widetilde{\gamma}\frac{\partial V}{\partial c} + \frac{1}{2}\sigma_c^2 c_0^2 \frac{\partial^2 V}{\partial c^2} + \frac{1}{2}\sigma_r^2 r_0 \frac{\partial^2 V}{\partial r^2} - r_0 V = 0.$$

Substituting the value of the parameters in this PDE leads to the following expression:

$$\frac{\partial V}{\partial t} - 0.0049\frac{\partial V}{\partial r} - 7.588\frac{\partial V}{\partial c} + 1695.9\frac{\partial^2 V}{\partial c^2} + +0.0008\frac{\partial^2 V}{\partial r^2} - 0.05V = 0.$$

For these parameter values, the boundary conditions for the green bond PDE (24) are as follows:

$$\begin{cases} V\left(T, r, c\right) = \left(c_T - K\right)^+ = \left(c_T - 50\right)^+, \\ V\left(t, r, c_{\min}\right) = 0, \\ V\left(t, r, c_{\max}\right) = \left(c_{\max} - Ke^{-r(T-t)}\right)^+ = 120 - 50e^{-r(1-t)}, \\ V\left(t, r_{\min}, c\right) = h_1(t, c), \\ V\left(t, r_{\max}, c\right) = h_2(t, c), \end{cases}$$

where h_1 can be obtained by solving the PDE below

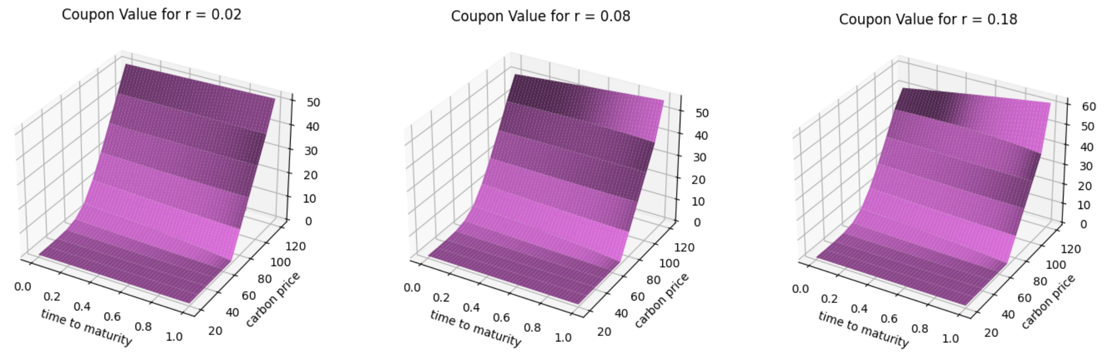
$$\begin{cases} \frac{\partial V}{\partial t} - 7.588\frac{\partial V}{\partial c} + 1695.9\frac{\partial^2 V}{\partial c^2} = 0, \\ V\left(t, r_{\min}, c_{\max}\right) = 0, \\ V\left(t, r_{\min}, c_{\max}\right) = \left(c_{\max} - Ke^{-r_{\min}(T-t)}\right)^+ = (120 - 50)^+ = 70, \\ V = \left(T, r_{\min}, c\right) = \left(c_T - K\right)^+ = \left(c_T - 50\right)^+. \end{cases}$$

Similarly, h_2 can be found by solving the following PDE:

$$\begin{cases} \frac{\partial V}{\partial t} - 7.588\frac{\partial V}{\partial c} + 1695.9\frac{\partial^2 V}{\partial c^2} - 0.20V = 0, \\ V\left(t, r_{\max}, c_{\min}\right) = 0, \\ V\left(t, r_{\max}, c_{\max}\right) = \left(c_{\max} - Ke^{-r_{\max}(T-t)}\right)^+ = 120 - 50e^{-0.20(1-t)}, \\ V = \left(T, r_{\max}, c\right) = \left(c_T - K\right)^+ = \left(c_T - 50\right)^+. \end{cases}$$

In the Python code, we observed that we can choose at most 50 grid points in the interest rate and carbon price direction, but for this choice, the code's running time is long. In Figures 29 and 30 below, we have that $\Delta r = 0.1$ and $\Delta c = 10$. Thus, we make 10 steps both in the interest rate and carbon price direction.

In Figure 29, the coupon values are given for three different interest rates: $r = 0.02$, $r = 0.08$, and $r = 0.18$.


Coupon Value for r = 0.02

Coupon Value for r = 0.08

Coupon Value for r = 0.18

(a) $r = 0.02$

(b) $r = 0.08$

(c) $r = 0.18$

Figure 29: Part 3: Coupon value for different values of r.

The value of the interest rate does not influence the shape of the plot of the coupon value. A reason for this can be found by closely examining the coefficients for the partial derivatives with respect to r in (35).

- $\frac{\partial V}{\partial r}$: $\alpha\left(\beta - r_0\right) - \lambda_r \sigma_r \sqrt{r_0}$,
- $\frac{\partial^2 V}{\partial r^2}$: $\frac{1}{2}\sigma_r^2 r_0$.

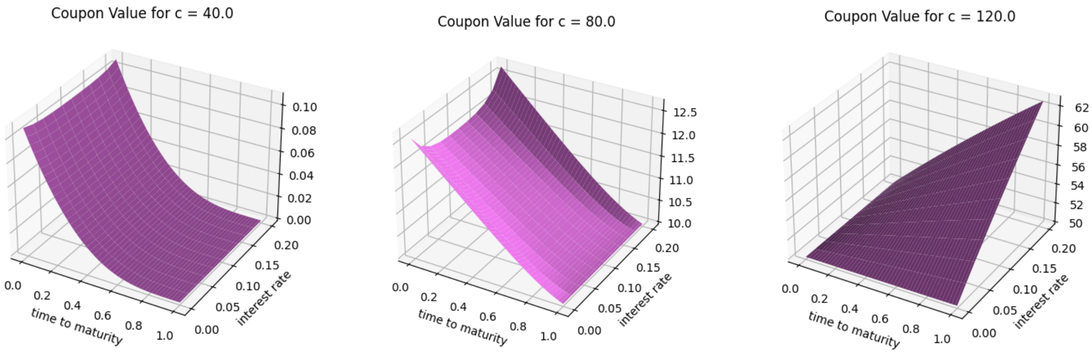
In Table 5, these coefficients are computed for the three different values of r_0 in Figure 29.

	$\alpha\left(\beta - r_0\right) - \lambda_r \sigma_r \sqrt{r_0}$	$\frac{1}{2}\sigma_r^2 r_0$
$r_0 = 0.02$	0.0226	$3.2041 \cdot 10^{-4}$
$r_0 = 0.08$	-0.0323	$1.2816 \cdot 10^{-3}$
$r_0 = 0.18$	-0.1235	$2.8837 \cdot 10^{-3}$

Table 5: Coefficients for partial derivatives w.r.t. r in (35).

The values of the coefficients are not heavily changed when r_0 changes. Therefore, the PDE stays almost the same and it is not surprising that the three plots in Figure 29 look very similar.

On the other hand, changing the value of the carbon price leads to different graphs for the coupon value. This is shown in Figure 30, where the carbon price c_0 is respectively 40, 80, and 120.



(a) $c = 40$ (b) $c = 80$ (c) $c = 120$

Figure 30: Part 3: Coupon value for different values of c.

To find a reason why the plots are so different, we again look at (35). In particular, we look at the coefficients of the partial derivatives w.r.t. the carbon price.

- $\frac{\partial V}{\partial c}$: $\mu c_0 - \lambda_c \sigma_c c_0$,
- $\frac{\partial^2 V}{\partial c^2}$: $\frac{1}{2}\sigma_c^2 c_0^2$.

In Table 6, these coefficients are computed for the three different values of r_0 in Figure 30.

	$\mu c_0 - \lambda_c \sigma_c c_0$	$\frac{1}{2}\sigma_c^2 c_0^2$
$c_0 = 40$	-4.336	665.6
$c_0 = 80$	-8.672	2666.2
$c_0 = 120$	-13.008	5990.4

Table 6: Coefficients for partial derivatives w.r.t. c in (35).

First of all, note that the values in Table 6 are much larger than the coefficients in Table 5. Modifying c_0 causes the PDE to change more than modifying r_0. Moreover, Table 6 shows that the values of the coefficients are heavily changed if c_0 is modified. Therefore, the PDE is changed and this leads to a different solution.

Without performing a Von Neumann analysis, we concluded in Chapter 4.6 that the numerical scheme is unstable, i.e., $|\xi| > 1$. Still, the coupon value graphs in Figure 29 and 30 look realistic, in the sense that we do not obtain Coupon values of order 10^{29}, such as in Figure 26a. This might be an indication that the amplification factor is not much bigger than 1. It can also be the case that the boundary conditions have a positive influence on the stability. This is something that a Von Neumann analysis does not take into account.

6 Conclusion and Discussion

6.1 Conclusion

The main aim of this master's thesis was to determine the stability of numerical schemes that are used for pricing green bonds. To this end, we consulted the master's thesis of Juriaan Rutten [29] to obtain a green bond model including a partial differential equation.

Before we were in the position to say something about stability, we started in Chapter 1 by presenting basic option valuation theory. Moreover, some mathematical preliminary definitions and theorems that can be used to derive a green bond model were shown. Then we shifted from pure mathematics to the subject of sustainable finance, which is very important in the battle against climate change. In Chapter 1.4, the notion of a green bond was introduced.

In Chapter 2, we focused again on mathematics. In particular, some basic notions for solving a PDE numerically were introduced. We dived into the theory of the finite difference method and time-stepping methods. Using the heat equation as an example, we learned how to derive a numerical scheme. We used both the inverse matrix and iterative solution methods for implicit schemes to make a step forward in time.

After Chapter 2, we gained enough knowledge to solve a PDE numerically, but this was not the goal of this master's project. The goal was to perform a stability analysis on a numerical scheme. Theory about this topic was introduced in Chapter 3, such as the amplification factor and Von Neumann stability. From here, we introduced the green bond model. After deriving multiple numerical schemes for this PDE in Chapter 4, we concluded that most schemes are unstable.

In Chapter 5, the numerical schemes were implemented in Python to check if the numerical results were correct. We concluded that some measurements can be taken to obtain reasonable results without having stability, such as choosing a small value of Δt and changing the values of some parameters to reduce the amplification factor.

6.2 Discussion

We already mentioned that the green bond market, and green bonds in general, is a relatively new subject. Therefore, there is much to discover. This master's thesis presents opportunities for further research. For instance, in Chapter 4.4, we assumed the carbon price to be constant, unlike the interest rate r. That is, we solved the PDE for multiple values of r. However, in the derivation of the numerical schemes, we chose r to be constant as well. This was done to make sure that we could perform a Von Neumann analysis. It might be interesting to check to what degree the stability gets better or worse if the value r is allowed to change. The same goes for the carbon price c in Chapter 4.5.

For this master's project, we derived multiple numerical schemes using forward and backward differences. We did not consider the two methods combined, i.e., the Crank-Nicolson method. Further research is required to conclude whether or not this method can be used to obtain stable numerical schemes. Moreover, in Chapter 5, we changed the value of some of the parameters, for example, Δc in Chapter 5.2. This was done to avoid computation errors, but this should be taken into account in the Von Neumann analysis in Chapter 4.5.5. The sine and cosine approximations become less accurate if we increase the value of Δc and future research could clarify to what degree the stability is changed.

Furthermore, in Chapter 4.6, we discretized a three-dimensional PDE, but the correlation coefficient ρ was assumed to be zero. This assumption caused the $\frac{\partial^2 V}{\partial c \partial r}$-term to vanish. In real life, this does not have to be the case. Further study could show how a nonzero ρ could influence the stability.

Finally, for this master project, we performed multiple Von Neumann analyses. This type of stability analysis does not take the boundary conditions or a final condition into account. It remains unclear what effect these conditions have on the stability of a numerical scheme. It is very worthwhile to dive into this topic.

References

[1] Desmond J Higham. *An introduction to financial option valuation: mathematics, stochastics and computation*, volume 13. Cambridge University Press, 2004.

[2] Cornelis W Oosterlee and Lech A Grzelak. *Mathematical modeling and computation in finance: with exercises and Python and MATLAB computer codes*. World Scientific, 2019.

[3] Hans Föllmer and Alexander Schied. *Stochastic finance: an introduction in discrete time*. Walter de Gruyter, 2011.

[4] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[5] Md Nurul Anwar and Laek Sazzad Andallah. A study on numerical solution of black-scholes model. *Journal of Mathematical Finance*, 8(2):372–381, 2018.

[6] Robert C Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973.

[7] Martin Bohner and Yao Zheng. On analytical solutions of the black–scholes equation. *Applied Mathematics Letters*, 22(3):309–313, 2009.

[8] The advantages and disadvantages of the black-scholes model. `https://www.smartcapitalmind.com/what-are-the-advantages-of-the-black-scholes-model.htm`, note = Accessed: 2010-09-30.

[9] Grigorios A Pavliotis. *Stochastic processes and applications*. Springer, 2016.

[10] Peter Mörters and Yuval Peres. *Brownian motion*, volume 30. Cambridge University Press, 2010.

[11] AG Malliaris. Wiener process. *Time Series and Statistics*, pages 316–318, 1990.

[12] Arthur Genthon. The concept of velocity in the history of brownian motion: From physics to mathematics and back. *The European Physical Journal H*, 45(1):49–105, 2020.

[13] PG Saffman and M Delbrück. Brownian motion in biological membranes. *Proceedings of the National Academy of Sciences*, 72(8):3111–3113, 1975.

[14] HC Brinkman. Brownian motion in a field of force and the diffusion theory of chemical reactions. *Physica*, 22(1-5):29–34, 1956.

[15] Frederik J Belinfante. On the mechanism of brownian motion in liquids. *American Journal of Physics*, 17(8):468–476, 1949.

[16] Steven E Shreve et al. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer, 2004.

[17] Wolfgang Arendt and Mahamadi Warma. Dirichlet and neumann boundary conditions: What is in between? *Nonlinear Evolution Equations and Related Topics: Dedicated to Philippe Bénilan*, pages 119–135, 2004.

[18] Brandon Washburn and DİK Mehmet. Derivation of black-scholes equation using itô's lemma. *Proceedings of International Mathematical Sciences*, 3(1):38–49, 2021.

[19] Fabrice Douglas Rouah. Four derivations of the black scholes pde. *Mathematical finance notes. http://www. frouah. com/pages/finmath. html. Accessed July*, 2020.

[20] Jeffrey D Sachs, Wing Thye Woo, Naoyuki Yoshino, and Farhad Taghizadeh-Hesary. Why is green finance important? 2019.

[21] Umair Saeed Bhutta, Adeel Tariq, Muhammad Farrukh, Ali Raza, and Muhammad Khalid Iqbal. Green bonds for sustainable development: Review of literature on development and impact of green bonds. *Technological Forecasting and Social Change*, 175:121378, 2022.

[22] Yves Steinebach, Xavier Fernández-i Marín, and Christian Aschenbrenner. Who puts a price on carbon, why and how? a global empirical analysis of carbon pricing policies. *Climate Policy*, 21(3):277–289, 2021.

[23] Gianfranco Gianfrate and Mattia Peri. The green advantage: Exploring the convenience of issuing green bonds. *Journal of cleaner production*, 219:127–135, 2019.

[24] Aaron Maltais and Björn Nykvist. Understanding the role of green bonds in advancing sustainability. *Journal of Sustainable Finance & Investment*, pages 1–20, 2020.

[25] SPINACI STEFANO. European green bonds: A standard for europe, open to the world. 2022.

[26] Allegra Pietsch and Dilyara Salakhova. Pricing of green bonds: drivers and dynamics of the greenium. 2022.

[27] European green bond standard (eugbs). `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32023R2631`, note = published 22 November 2023.

[28] Andreas Karpf and Antoine Mandel. Does it pay to be green? *Available at SSRN 2923484*, 2017.

[29] Jurriaan Rutten. Green bond valuation: A numerical mathematics perspective: Assessing the influence of environmental factors. 2024.

[30] Martin Braun and Martin Golubitsky. *Differential equations and their applications*, volume 2. Springer, 1983.

[31] C Vuik and DJP Lahaye. Scientific computing (wi4201). *Lecture notes for wi4201*, 2012.

[32] Darae Jeong, Minhyun Yoo, and Junseok Kim. Finite difference method for the black–scholes equation without boundary conditions. *Computational Economics*, 51:961–972, 2018.

[33] Cornelis Vuik, FJ Vermolen, MB van Gijzen, and Thea Vuik. Numerical methods for ordinary differential equations. 2023.

[34] Leonhard Euler. Institutionum calculi integralis volumen tertium. 1770.

[35] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical proceedings of the Cambridge philosophical society*, volume 43, pages 50–67. Cambridge University Press, 1947.

[36] Charles L Byrne. *Applied iterative methods*. AK Peters Wellesley, 2008.

[37] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[38] Henk A Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.

[39] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[40] Christopher C Paige and Michael A Saunders. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis*, 12(4):617–629, 1975.

[41] Stanislaw Ulam, Harold William Kuhn, Albert William Tucker, and Claude E Shannon. John von neumann, 1903-1957. In *The Intellectual Migration: Europe and America, 1930-1960*, pages 235–269. Harvard University Press, 1969.

[42] Jules G Charney, Ragnar Fjörtoft, and J von Neumann. Numerical integration of the barotropic vorticity equation. *Tellus*, 2(4):237–254, 1950.

[43] Heath Windcliff, Peter A Forsyth, and Ken R Vetzal. Analysis of the stability of the linear boundary condition for the black-scholes equation. *Journal of Computational Finance*, 8:65–92, 2004.

A Equivalence of the Black-Scholes PDE and the heat equation

Theorem 13. *The Black-Scholes partial differential equation, given by*

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{37}$$

for $V(S,t)$ is equivalent to the heat equation, given by

$$\frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2} \tag{38}$$

for $y(x,\tau)$.

Proof. To do this, we proceed as follows. Write $\tau(t) = T - \frac{1}{2}\sigma^2 t$, where T is the maturity time. Then by the chain rule, we find that

$$\frac{\partial V}{\partial t} = \frac{\partial V}{\partial \tau}\frac{\partial \tau}{\partial t} = \frac{\partial V}{\partial \tau} \cdot -\frac{1}{2}\sigma^2 = -\frac{1}{2}\sigma^2 \frac{\partial V}{\partial \tau}. \tag{39}$$

Moreover, if we use the transformation $S = Ee^x$, we find that

$$S(x) = Ee^x \iff x(S) = \ln\left(\frac{1}{E}S\right) = \ln\left(\frac{1}{E}\right) + \ln(S).$$

Now we are in the position to compute both the first and second-order derivative of x with respect to S:

$$\frac{\partial x}{\partial S} = \frac{1}{S} \text{ and } \frac{\partial^2 x}{\partial S^2} = -\frac{1}{S^2}.$$

The next step is to derive expressions for $\frac{\partial V}{\partial S}$ and $\frac{\partial^2 V}{\partial S^2}$ using the above obtained results. We start with $\frac{\partial V}{\partial S}$:

$$\frac{\partial V}{\partial S} = \frac{\partial V}{\partial x}\frac{\partial x}{\partial S} = \frac{1}{S}\frac{\partial V}{\partial x}, \tag{40}$$

and continue with $\frac{\partial^2 V}{\partial S^2}$, where we use both the product rule and the chain rule:

$$\begin{aligned}\frac{\partial^2 V}{\partial S^2} &= \frac{\partial^2 x}{\partial S^2}\frac{\partial V}{\partial x} + \frac{\partial^2 V}{\partial x^2}\left(\frac{\partial x}{\partial S}\right)^2 \\ &= -\frac{1}{S^2}\frac{\partial V}{\partial x} + \frac{1}{S^2}\frac{\partial^2 V}{\partial x^2}. \end{aligned}\tag{41}$$

If we substitute (39), (40), and (41) into the Black-Scholes equation (37), we obtain the following:

$$-\frac{1}{2}\sigma^2\frac{\partial V}{\partial \tau} + \frac{1}{2}\sigma^2 S^2\left(-\frac{1}{S^2}\frac{\partial V}{\partial x} + \frac{1}{S^2}\frac{\partial^2 V}{\partial x^2}\right) + rS\frac{1}{S}\frac{\partial V}{\partial x} - rV = 0.$$

Multiplying both sides with -1 and dividing by $\frac{1}{2}\sigma^2$ leads to the next PDE:

$$\frac{\partial V}{\partial \tau} - \frac{\partial^2 V}{\partial x^2} + \left(1 - \frac{2r}{\sigma^2}\right)\frac{\partial V}{\partial x} + \frac{2r}{\sigma^2}V = 0. \tag{42}$$

For the next step, let $V = Ee^{\gamma x + \delta \tau} y(x, \tau)$. With this notation, we obtain expressions for $\frac{\partial V}{\partial \tau}$, $\frac{\partial V}{\partial x}$, and $\frac{\partial^2 V}{\partial x^2}$. For the next calculations, we again use the product rule for differentiation multiple times.

$$\frac{\partial V}{\partial \tau} = E\delta e^{\gamma x + \delta \tau} y(x, \tau) + Ee^{\gamma x + \delta \tau}\frac{\partial y}{\partial \tau}. \tag{43}$$

$$\frac{\partial V}{\partial x} = E\gamma e^{\gamma x + \delta \tau} y(x, \tau) + Ee^{\gamma x + \delta \tau}\frac{\partial y}{\partial x}. \tag{44}$$

$$\begin{aligned}\frac{\partial^2 V}{\partial x^2} &= E\gamma^2 e^{\gamma x + \delta \tau} y(x, \tau) + E\gamma e^{\gamma x + \delta \tau}\frac{\partial y}{\partial x} + Ee^{\gamma x + \delta \tau}\frac{\partial^2 y}{\partial x^2} + E\gamma e^{\gamma x + \delta \tau}\frac{\partial y}{\partial x} \\ &= E\gamma^2 e^{\gamma x + \delta \tau} y(x, \tau) + 2E\gamma e^{\gamma x + \delta \tau}\frac{\partial y}{\partial x} + Ee^{\gamma x + \delta \tau}\frac{\partial^2 y}{\partial x^2}. \end{aligned} \tag{45}$$

By choosing

- $\gamma = \frac{\sigma^2 - 2r}{2\sigma^2}$ and
- $\delta = -\left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2$,

and substituting (43), (44), and (45) into equation (42), we divide by $Ee^{\gamma x + \delta \tau}$, which is not equal to 0, to find:

$$\delta y(x, \tau) + \frac{\partial y}{\partial \tau} - \gamma^2 y(x, \tau) - \cancel{2\gamma\frac{\partial y}{\partial x}} - \frac{\partial^2 y}{\partial x^2} + 2\gamma^2 y(x, \tau) + \cancel{2\gamma\frac{\partial y}{\partial x}} + (1 - 2\gamma)y(x, \tau) = 0.$$

After collecting the terms, observe that the obtained PDE already looks very similar to the heat equation:

$$\left(\delta + \gamma^2 - 2\gamma + 1\right) y(x, \tau) + \frac{\partial y}{\partial \tau} - \frac{\partial^2 y}{\partial x^2} = 0 \tag{46}$$

It suffices to show that the $y(x, \tau)$-terms cancel each other out. That is:

$$\begin{aligned}\delta + \gamma^2 - 2\gamma + 1 &= \delta + (\gamma - 1)^2 \\ &= -\left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2 + \left(\frac{\sigma^2 - 2r}{2\sigma^2} - 1\right)^2 \\ &= -\left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2 + \left(\frac{-\sigma^2 - 2r}{2\sigma^2}\right)^2 \\ &= -\left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2 + \left(\frac{\sigma^2 + 2r}{2\sigma^2}\right)^2 \\ &= 0.\end{aligned}$$

Since there is no $y(x, \tau)$-term in equation (46), we have arrived at the heat equation by taking the second-order derivative of y with respect to x in equation (46) to the right-hand side. □

B Appendix: Python code

B.1 Code for the 3-dimensional green bond PDE

```
# Matlab Program: Green PDE (3 dimensional).
# Explicit method (backward differences).

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

# Gamma and gammatilde: abbrivitions that are also used in the thesis.
def GAMMA(alpha, beta, r_0, lambda_r, sigma_r):
    gamma = alpha*(beta - r_0) - lambda_r * sigma_r * np.sqrt(r_0)
    return gamma

def GAMMATILDE(mu, c_0, lambda_c, sigma_c):
    gammatilde = mu*c_0 - lambda_c*sigma_c*c_0
    return gammatilde

# Coefficients for the numerical schemes.
def COEF1(sigma_r, r_0, dt, dr, gamma):
    return sigma_r**2*r_0*dt/(2*dr**2) - gamma*dt/dr

def COEF2(sigma_c, c_0, dt, dc, gammatilde):
    return sigma_c**2*c_0**2*dt/(2*dc**2) - gammatilde*dt/dc

def COEF3(gammatilde, dt, dc, dr, sigma_c, c_0, sigma_r, r_0):
    coef3 = (1 + gammatilde*dt/dc + gamma*dt/dr -
             sigma_c**2*c_0**2*dt/(dc**2) -
             sigma_r**2*r_0*dt/(dr**2) - r_0*dt)
    return coef3

def COEF4(sigma_r, r_0, dt, dr):
    return sigma_r**2*r_0*dt/(2*dr**2)

def COEF5(sigma_c, c_0, dt, dc):
    return sigma_c**2*c_0**2*dt/(2*dc**2)

# Coefficients for the numerical schemes for the functions h_1 and h_2.
def h_COEF1(sigma_c, c_0, dt, dc, gammatilde):
    return (sigma_c ** 2 * c_0 ** 2 * dt / (2 * dc ** 2) -
            gammatilde * dt / dc)

def h_COEF2(gammatilde, dt, dc, sigma_c, c_0, r_0):
    return (1 + gammatilde * dt / dc -
            sigma_c ** 2 * c_0 ** 2 * dt / (dc ** 2) - r_0 * dt)

def h_COEF3(dt, dc, sigma_c, c_0):
    return sigma_c ** 2 * c_0 ** 2 * dt / (2 * dc ** 2)

# Initialisation of the parameters.
alpha = 0.91
beta = 0.0451
mu = 0.058
t_min = 0
t_max = 1
r_min = 0
r_max = 0.20
c_min = 20
c_max = 120
r_0 = 0.05
c_0 = 0.5*(c_max + c_min)
lambda_r = 0.01
lambda_c = 0.2
sigma_r = 0.179
sigma_c = 0.2
gamma = GAMMA(alpha, beta, r_0, lambda_r, sigma_r)
gammatilde = GAMMATILDE(mu, c_0, lambda_c, sigma_c)
T = 1.0 # Time indicator (t_max - t_min = 1-0=1).
K = 70 # Strike price.

# Step sizes
maxt = 1000 # Number of time steps.
maxr = 10 # Number of interest steps, upper limit: 50.
maxc = 10 # Number of carbon price steps, upper limit: 50.
t_length = t_max-t_min
r_length = r_max-r_min
c_length = c_max-c_min
dt = t_length/maxt
dr = r_length/maxr
dc = c_length/maxc

# Python initialisation
time = np.zeros(maxt+1) # Time to maturity.
r = np.zeros(maxr+1)
c = np.zeros(maxc+1)


V = np.zeros([maxt+1, maxr+1, maxc+1])
for i in range(maxt+1):
    time[i] = t_min + i*dt
```

```
for i in range(maxr+1):
    r[i] = r_min + i*dr

for i in range(maxc+1):
    c[i] = c_min + i*dc

# Final condition.
# V_i(T_i,r,c) = (c_T_{i} - K_i)^{+}.
for i in range(maxr+1):
    for j in range(maxc+1):
        V[:,i,j] = np.maximum(c[j]-K,0)

# Boundary conditions.
# V_i(t,r,c_min) = 0.
V[:,:,0] = 0

# V_i(t,r,c_max) = (c_max - K_i*e^{-r*t})^{+}.
for i in range(maxr+1):
    for j in range(maxt+1):
        V[j,i,maxc] = np.maximum(c_max-K*np.exp(-r[i]*time[j]),0)

# The functions h_1 and h_2 will also be computed
# numerically using an explicit method.
# The numerical scheme is derived in case 2(b) (Backward Differences).
# V_i(t,r_min,c) = h_1(t,c)
h1_coef1 = h_COEF1(sigma_c, c_0, dt, dc, gammatilde)
h1_coef2 = h_COEF2(gammatilde, dt, dc, sigma_c, c_0, r_min)
h1_coef3 = h_COEF3(dt, dc, sigma_c, c_0)

# Implementation of the explicit method
for i in range(maxt): # Time loop
    for j in range(1,maxc): # Carbon price loop
        V[maxt-i-1,0,j] = (h1_coef1*V[maxt-i,0,j-1] +
                           h1_coef2*V[maxt-i,0,j] +
                           h1_coef3*V[maxt-i,0,j+1])

# V_i(t,r_max,c) = h_2(t,c)
h2_coef1 = h_COEF1(sigma_c, c_0, dt, dc, gammatilde)
h2_coef2 = h_COEF2(gammatilde, dt, dc, sigma_c, c_0, r_max)
h2_coef3 = h_COEF3(dt, dc, sigma_c, c_0)

# Implementation of the explicit method
for i in range(maxt): # Time loop
    for j in range(1,maxc): # Carbon price loop
        V[maxt-i-1,maxr,j] = (h1_coef1*V[maxt-i,maxr,j-1] +
                                h1_coef2*V[maxt-i,maxr,j] +
                               h1_coef3*V[maxt-i,maxr,j+1])

# Now we will compute the solution of the original green PDE.
coef1 = COEF1(sigma_r, r_0, dt, dr, gamma)
coef2 = COEF2(sigma_c, c_0, dt, dc, gammatilde)
coef3 = COEF3(gammatilde, dt, dc, dr, sigma_c, c_0, sigma_r, r_0)
coef4 = COEF4(sigma_r, r_0, dt, dr)
coef5 = COEF5(sigma_c, c_0, dt, dc)

for i in range(maxt):
    for j in range(1,maxr):
        for k in range(1,maxc):
            V[maxt-1-i,j,k] = (coef1*V[maxt-i,j-1,k] +
                                coef2*V[maxt-i,j,k-1] +
                                coef3*V[maxt-i,j,k] +
                                coef4*V[maxt-i,j+1,k] +
                                coef5*V[maxt-i,j,k+1])

#Generating a plot
timeMatrix = np.zeros([maxt+1, maxc+1])
interestMatrix = np.zeros([maxt+1,maxr+1])
carbonMatrix = np.zeros([maxt+1,maxc+1])

for i in range(maxt+1):
    timeMatrix[i,:] = time[i]

for i in range(maxr+1):
    interestMatrix[:,i] = r[i]

for i in range(maxc+1):
    carbonMatrix[:,i] = c[i]

# Plot the surface for a fixed r.
fig = plt.figure(1)
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, carbonMatrix, V[:,3,:], color=[1,0.5,1])
plt.xlabel('time␣to␣maturity')
plt.ylabel('carbon␣price')
plt.title('Coupon␣Value␣for␣r␣=␣'+ str(r[3]))
plt.show()

print(r)
print(c)
# Plot the surface for a fixed c.
fig = plt.figure(2)
```

```
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, interestMatrix, V[:,:,10], color=[1,0.5,1])
plt.xlabel('time to maturity')
plt.ylabel('interest rate')
plt.title('Coupon Value for c = '+ str(c[10]))
plt.show()
```

B.2 Code for Part 1(a): c is constant, r is not: forward differences

```
# Matlab Program: Part 1(a)
# Implicit Method

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from scipy.sparse.linalg import bicgstab
from scipy.sparse.linalg import gmres

def GAMMA(alpha, beta, r_0, lambda_r, sigma_r):
    gamma = (alpha*(beta - r_0) -
             lambda_r * sigma_r * np.sqrt(r_0))
    return gamma

def COEF1(dt, dr, sigma_r, r_0):
    coef1 = -sigma_r**2*r_0*dt/(2*dr**2)
    return coef1

def COEF2(gamma,dt,dr,sigma_r,r_0):
    coef2 = (1 + gamma*dt/dr
             + sigma_r**2*r_0*dt/(dr**2) + r_0*dt)
    return coef2

def COEF3(gamma, dt, dr, sigma_r, r_0):
    coef3 = (-gamma*dt/dr -
             sigma_r**2*r_0*dt/(2*dr**2))
    return coef3

# Initialisation of the parameters.
alpha = 0.91
beta = 0.0451
t_min = 0
t_max = 1
r_min = 0
r_max = 0.20
r_0 = 0.05
c_0 = 70
lambda_r = 0.01
sigma_r = 0.179
gamma = GAMMA(alpha, beta, r_0, lambda_r, sigma_r)
T = 1.0 # Time indicator (t_max - t_min = 1-0=1).
K = 50 # Strike price.

# Step sizes
```

```
maxt = 1000 # Number of time steps.
maxr = 100 # Number of interest steps.
t_length = t_max-t_min
r_length = r_max-r_min
dt = t_length/maxt
dr = r_length/maxr

# Python initialisation
time = np.zeros(maxt+1) # Time to maturity.
r = np.zeros(maxr+1)
V = np.zeros([maxt+1, maxr+1])

for i in range(maxt+1):
    time[i] = t_min + i*dt

for i in range(maxr+1):
    r[i] = r_min + i*dr

# Boundary conditions:
# V(t,r_min) = (c_0 - K*e^{-r_min*t})^{+}
# V(t,r_max) = (c_0 - K*e^{-r_max*t})^{+}
for i in range(maxt+1):
    V[i,0] = np.maximum(c_0 - K*np.exp(-r[0]*time[i]),0)
    V[i,maxr] = np.maximum(c_0 -
                            K*np.exp(-r[maxr]*time[i]),0)

# Final condition:
# V(T,r) = (c_0 - K)^{+}.
V[0,:] = np.maximum(c_0-K, 0)

coef1 = COEF1(dt, dr, sigma_r, r_0)
coef2 = COEF2(gamma,dt,dr,sigma_r,r_0)
coef3 = COEF3(gamma,dt,dr,sigma_r,r_0)

# Implementation of the implicit method
B = np.zeros([maxr-1,maxr-1])
for i in range(maxr-1):
    B[i,i] = coef2
    if i != maxr-2:
        B[i+1,i] = coef1
        B[i,i+1] = coef3

B_inv = np.linalg.inv(B)
u = np.zeros(maxr-1)
# Implementation of the implicit method
for k in range(1,maxt+1): # Time Loop
```

```
    u[0] = coef1*V[k,0]
    u[maxr-2] = coef3*V[k,maxr]
    # # Inverse matrix
    # V[k,1:maxr] = np.matmul(B_inv, V[k-1,1:maxr] - u)
    # # Bi-CGSTAB:
    V[k,1:maxr], exitcode = bicgstab(B, V[k-1,1:maxr] - u, x0=None, atol=0.0)
    # # GMRES:
    # V[k, 1:maxr], exitcode = gmres(B, V[k-1,1:maxr] - u, x0=None, atol=0.0)

# print(V)

#Generating a plot
timeMatrix = np.zeros([maxt+1, maxr+1])
interestMatrix = np.zeros([maxt+1,maxr+1])

for i in range(maxt+1):
    timeMatrix[i,:] = time[i]

for i in range(maxr+1):
    interestMatrix[:,i] = r[i]

# Plot the surface for a fixed c.
fig = plt.figure(1)
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, interestMatrix, V, color=[1,0.5,1])
plt.xlabel('time␣to␣maturity')
plt.ylabel('interest␣rate')
plt.title('Coupon␣Value␣for␣c␣=␣'+ str(c_0))
plt.show()
```

B.3 Code for Part 1(b): c is constant, r is not: backward differences

```
# Matlab Program: Part 1(b)
# Explicit Method

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

def GAMMA(alpha, beta, r_0, lambda_r, sigma_r):
    gamma = (alpha*(beta - r_0) -
             lambda_r * sigma_r * np.sqrt(r_0))
    return gamma

def COEF1(gamma, dt, dr, sigma_r, r_0):
    coef1 = (-gamma*dt/dr +
             sigma_r**2*r_0*dt/(2*dr**2))
    return coef1

def COEF2(gamma,dt,dr,sigma_r,r_0):
    coef2 = (1 + gamma*dt/dr
             - sigma_r**2*r_0*dt/(dr**2) - r_0*dt)
    return coef2

def COEF3(dt, dr, sigma_r, r_0):
    coef3 = sigma_r**2*r_0*dt/(2*dr**2)
    return coef3

# Initialisation of the parameters.
alpha = 0.91
beta = 0.0451
t_min = 0
t_max = 1
r_min = 0
r_max = 0.20
r_0 = 0.05
c_0 = 70
lambda_r = 0.01
sigma_r = 0.179
gamma = GAMMA(alpha, beta, r_0, lambda_r, sigma_r)
T = 1.0 # Time indicator (t_max - t_min = 1-0=1).
K = 50 # Strike price.

# Step sizes
maxt = 1000 # Number of time steps.
maxr = 100 # Number of interest steps.
```

```
t_length = t_max-t_min
r_length = r_max-r_min
dt = t_length/maxt
dr = r_length/maxr

# Python initialisation
time = np.zeros(maxt+1) # Time to maturity.
r = np.zeros(maxr+1)
V = np.zeros([maxt+1, maxr+1])

for i in range(maxt+1):
    time[i] = t_min + i*dt

for i in range(maxr+1):
    r[i] = r_min + i*dr

# Boundary conditions:
# V(t,r_min) = (c_0 - K*e^{-r_min*(T-t)})^{+}.
# V(t,r_max) = (c_0 - K*e^{-r_max*(T-t)})^{+}.
for i in range(maxt+1):
    V[i,0] = np.maximum(c_0 - K*np.exp(-r[0]*time[i]),0)
    V[i,maxr] = np.maximum(c_0 -
                            K*np.exp(-r[maxr]*time[i]),0)

# Final condition:
# V(T,r) = (c_0 - K)^{+}.
V[0,:] = np.maximum(c_0-K, 0)

coef1 = COEF1(gamma, dt, dr, sigma_r, r_0)
coef2 = COEF2(gamma,dt,dr,sigma_r,r_0)
coef3 = COEF3(dt, dr, sigma_r, r_0)

# Implementation of the explicit method
for i in range(1,maxt+1): # Time loop
    for j in range(1,maxr): # Interest rate loop
        V[i,j] = (coef1*V[i-1,j-1] + coef2*V[i-1,j]
                  + coef3*V[i-1,j+1])

# print(V)

#Generating a plot
timeMatrix = np.zeros([maxt+1, maxr+1])
interestMatrix = np.zeros([maxt+1,maxr+1])

for i in range(maxt+1):
    timeMatrix[i,:] = time[i]
```

```
for i in range(maxr+1):
    interestMatrix[:,i] = r[i]

# Plot the surface for a fixed c.
fig = plt.figure(1)
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, interestMatrix, V, color=[1,0.5,1])
plt.xlabel('time to maturity')
plt.ylabel('interest rate')
plt.title('Coupon Value for c = '+ str(c_0))
plt.show()
```

B.4 Code for Part 2(a): r is constant, c is not: forward differences

```
# Matlab Program: Part 2(a)
# Implicit Method

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from scipy.sparse.linalg import bicgstab
from scipy.sparse.linalg import gmres

def GAMMATILDE(mu, c_0, lambda_c, sigma_c):
    gammatilde = mu*c_0 - lambda_c*sigma_c*c_0
    return gammatilde

def COEF1(sigma_c, c_0, dt, dc, gammatilde):
    coef1 = sigma_c**2*c_0**2*dt/(2*dc**2) - gammatilde*dt/dc
    return coef1

def COEF2(gammatilde, dt, dc, sigma_c, c_0, r_0):
    coef2 = 1 + gammatilde*dt/dc - sigma_c**2*c_0**2*dt/(dc**2) - r_0*dt
    return coef2

def COEF3(dt, dc, sigma_c, c_0):
    coef3 = sigma_c**2*c_0**2*dt/(2*dc**2)
    return coef3

# Initialisation of the parameters.
mu = 0.058
t_min = 0
t_max = 1
c_min = 20
c_max = 120
r_0 = 0.05
c_0 = 7
lambda_c = 0.2
sigma_c = 0.832
gammatilde = GAMMATILDE(mu, c_0, lambda_c, sigma_c)
T = 1.0 # Time indicator (t_max - t_min = 1-0=1).
K = 5 # Strike price.

# Step sizes
maxt = 1000 # Number of time steps.
maxc = 10 # Number of carbon price steps.
t_length = t_max-t_min
c_length = c_max-c_min
```

```
dt = t_length/maxt
dc = c_length/maxc

# Python initialisation
time = np.zeros(maxt+1) # Time to maturity.
c = np.zeros(maxc+1)
V = np.zeros([maxt+1, maxc+1])

for i in range(maxt+1):
    time[i] = t_min + i*dt

for i in range(maxc+1):
    c[i] = c_min + i*dc

# Boundary condition:
# V(t,c_min) = 0.
# V(t,c_max) = (c_max - K*e^{-rt})^{+}
for i in range(maxt+1):
    V[i,0] = 0
    V[i,maxc] = np.maximum(c[maxc] -
                           K*np.exp(-r_0*time[i]),0)

# Final condition:
# V(T,c) = (c_T - K)^{+}.
for j in range(maxc+1):
    V[0,j] = np.maximum(c[j]-K,0)

coef1 = COEF1(sigma_c, c_0, dt, dc, gammatilde)
coef2 = COEF2(gammatilde, dt, dc, sigma_c, c_0, r_0)
coef3 = COEF3(dt, dc, sigma_c, c_0)

# Implementation of the implicit method via
# an inverse matrix.
B = np.zeros([maxc-1,maxc-1])
for i in range(maxc-1):
    B[i,i] = coef2
    if i != maxc-2:
        B[i+1,i] = coef1
        B[i,i+1] = coef3

B_inv = np.linalg.inv(B)
u = np.zeros(maxc-1)
# Implementation of the implicit method
for k in range(1,maxt+1): # Time Loop
    u[0] = coef1*V[k,0]
    u[maxc-2] = coef3*V[k,maxc]
    # # Inverse matrix
    V[k,1:maxc] = np.matmul(B_inv, V[k-1,1:maxc] - u)
    # # Bi-CGSTAB:
    # V[k,1:maxc], exitcode = bicgstab(B, V[k-1,1:maxc] - u, x0=None, atol=0.0)
    # # GMRES:
    # V[k, 1:maxc], exitcode = gmres(B, V[k-1,1:maxc] - u, x0=None, atol=0.0)

# print(V)

#Generating a plot
timeMatrix = np.zeros([maxt+1, maxc+1])
carbonMatrix = np.zeros([maxt+1,maxc+1])

for i in range(maxt+1):
    timeMatrix[i,:] = time[i]

for i in range(maxc+1):
    carbonMatrix[:,i] = c[i]

# Plot the surface for a fixed r.
fig = plt.figure(1)
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, carbonMatrix, V, color=[1,0.5,1])
plt.xlabel('time to maturity')
plt.ylabel('carbon price')
plt.title('Coupon Value for r = '+ str(r_0))
plt.show()
```

B.5 Code for Part 2(b): r is constant, c is not: backward differences

```
# Matlab Program: Part 2(b)
# Explicit Method

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

def GAMMATILDE(mu, c_0, lambda_c, sigma_c):
    gammatilde = mu*c_0 - lambda_c*sigma_c*c_0
    return gammatilde

def COEF1(sigma_c, c_0, dt, dc, gammatilde):
    coef1 = sigma_c**2*c_0**2*dt/(2*dc**2) - gammatilde*dt/dc
    return coef1

def COEF2(gammatilde, dt, dc, sigma_c, c_0, r_0):
    coef2 = 1 + gammatilde*dt/dc - sigma_c**2*c_0**2*dt/(dc**2) - r_0*dt
    return coef2

def COEF3(dt, dc, sigma_c, c_0):
    coef3 = sigma_c**2*c_0**2*dt/(2*dc**2)
    return coef3

# Initialisation of the parameters.
mu = 0.058
t_min = 0
t_max = 1
c_min = 20
c_max = 120
r_0 = 0.05
c_0 = 70
lambda_c = 0.2
sigma_c = 0.832
gammatilde = GAMMATILDE(mu, c_0, lambda_c, sigma_c)
T = 1.0 # Time indicator (t_max - t_min = 1-0=1).
K = 50 # Strike price.

# Step sizes
maxt = 1000 # Number of time steps.
maxc = 10 # Number of carbon price steps.
t_length = t_max-t_min
c_length = c_max-c_min
dt = t_length/maxt
dc = c_length/maxc
```

```
# Python initialisation
time = np.zeros(maxt+1) # Time to maturity.
c = np.zeros(maxc+1)
V = np.zeros([maxt+1, maxc+1])

for i in range(maxt+1):
    time[i] = t_min + i*dt

for i in range(maxc+1):
    c[i] = c_min + i*dc

# Boundary condition:
# V(t,c_min) = 0.
# V(t,c_max) = (c_max - K*e^{-r(T-t)})^{+}.
for i in range(maxt+1):
    V[i,0] = 0
    V[i,maxc] = np.maximum(c[maxc] -
                           K*np.exp(-r_0*time[i]),0)

# Final condition:
# V(T,c) = (c_T - K)^{+}.
for j in range(maxc+1):
    V[0,j] = np.maximum(c[j]-K,0)

coef1 = COEF1(sigma_c, c_0, dt, dc, gammatilde)
coef2 = COEF2(gammatilde, dt, dc, sigma_c, c_0, r_0)
coef3 = COEF3(dt, dc, sigma_c, c_0)

# Implementation of the explicit method
for i in range(1,maxt+1): # Time loop
    for j in range(1,maxc): # Interest rate loop
        V[i,j] = (coef1*V[i-1,j-1] + coef2*V[i-1,j]
                    + coef3*V[i-1,j+1])

# print(V)

#Generating a plot
timeMatrix = np.zeros([maxt+1, maxc+1])
carbonMatrix = np.zeros([maxt+1,maxc+1])

for i in range(maxt+1):
    timeMatrix[i,:] = time[i]

for i in range(maxc+1):
    carbonMatrix[:,i] = c[i]
```

```
# Plot the surface for a fixed r.
fig = plt.figure(1)
ax = fig.gca(projection='3d')
ax.plot_surface(timeMatrix, carbonMatrix, V, color=[1,0.5,1])
plt.xlabel('time to maturity')
plt.ylabel('carbon price')
plt.title('Coupon Value for r = '+ str(r_0))
plt.show()
```