Profitable disasters: Modelling a CAT bond using barrier options

Flore Steenberghe

Department of Mathematics
Bachelor Thesis
Wiskunde en Toepassingen
Thesis advisor: Prof. dr. ir. C.W. Oosterlee
June 17, 2024

1 Abstract

Catastrophic events can cause significant damage. The damages caused by these extreme events can be so costly that insurance companies cannot cover all of them. To bridge this gap and limit financial risk, insurance companies (or other corporations or agencies that might need to cover this gap) can use CAT (catastrophic) bonds. There are several triggers or thresholds that can determine the payout of a CAT bond. We will specifically look at parametric triggers, where the issuer of the bond can collect after a specific trigger surpasses a threshold (Polack, 2018).
To represent this threshold we can use barrier options. In this paper, we will find the value of the down-and-in barrier put option for a specific catastrophic event. For this event, we looked at the gas fields in Groningen. The data analysed in the paper by Pijpers is the temporal distribution of the earthquakes, that is the time interval between earthquakes with a magnitude grater or equal to 1.2. He found that the Weibull distribution fits best for this data (Pijpers, 2018). We have used this Weibull distribution to represent the earthquakes in Groningen.
To mimic a CAT bond we say the following: Suppose a house in Groningen, located in the gas fields, is valued at 300,000 euros, and we want to keep the value of this house. If the value of the house were to go down to a barrier B, the CAT bond will be paid out and we will look at the value of this down-and-in barrier put option.

To find the value of this put option we will look further into CAT bonds and barrier options. After this, the properties of the Weibull distribution are examined, together with the parameters we want to use. These are the parameters found by Pijpers in the zone South-East of the gas fields. To find these parameters he used the maximum likelihood estimator.
To calculate the value of options, we can use the COS method. First, the COS method to approximate probability density functions will be explained. After this, we will look at the COS method for the European call option, and finally, we will examine the COS method for discretely-monitored barrier options.
Once we have combined all of this information, we will be able to compute a catastrophic bond, with parametric trigger earthquakes, using a down-and-in barrier put option.

Contents

2 CAT bonds and barrier options

2.1 CAT bonds

In the summery for policymakers, the IPCC have found that climate extremes will be more and more common in the future with the rise of the earth temperature. "Heavy rain fall, heatwaves and an increase in floods" (Intergovernmental Panel on Climate Change, 2021. Statement B.2). As one can imagine, some of these will lead to too much damage combined with costs of millions of dollars. The increase of climate extremes causes an increase of financial pressure on insurance companies. To enhance financial protection, these companies have developed CAT (catastrophic) bonds. Since the mid 90's, insurance companies have used CAT bonds to limit their financial risk. Not only insurance companies, but also reinsurance companies, counties, countries, non-profit organisations and corporations can sponsor CAT bonds. For example, corporations like Google issues CAT bonds for practical reasons. Their headquarters is located in California, an area facing many earthquakes. Therefore, they could face substantial losses. These institutions issue the bonds to the investor market, they transfer the financial risk to the financial market (Niki, 2023).

To isolate financial risk, an issuer creates a special purpose vehicle, SPV. A SPV is a secondary separate company. Because the organisations are separate, the legal status guarantees its liability. In case the parent company should go bankrupt, the SPV wouldn't necessarily go bankrupt too (Hayes, 2022). That is why using an SPV is a good option to distribute the risk.
The issuer of the bond and the SPV agree upon a threshold or trigger for an extreme event and a duration of the bond. During the duration of the bond, the issuer pays the SPV a premium. On the other end we have investors or hedge funds that can put money into the bonds, and in return for this they receive a coupon. The SPV collects this principal of the investors and the premiums of the issuers, and invests these into low-risk accounts. As a result, the value of the money is increased and the coupon of the investors can be payed. If the contractually agreed-upon threshold is reached, the issuer can collect and the investor looses all its principle. If the issuer doesn't collect, the principle is returned as well (after the contract duration). In Figure 1 you can see the CAT bond structure.
There are several different thresholds or triggers that are used for CAT bonds. The most common are indemnity, industry loss and parametric triggers. For indemnity triggers, the payout will be the insurance losses of the issuer (Polack). It will take a while before all the losses of the issuer are finalised, hence the payout process will take a while. The payout of industry loss triggers are based on what the insurance industry losses. After the so-called attachment point, which is set in the contract, the issuers will be paid (Niki). This will also take a long time, since it needs a third party to provide an independent estimate of the losses (Polack). The parametric triggers will be paid when a certain trigger surpasses a threshold. An example would be a payout after an earthquake greater than 7.0 on the Richter Scale. This trigger is the easy to verify, since it is determined by a measurable quantity.
Even tough a CAT bond carries the risk for an investor to lose all its principles, the bonds

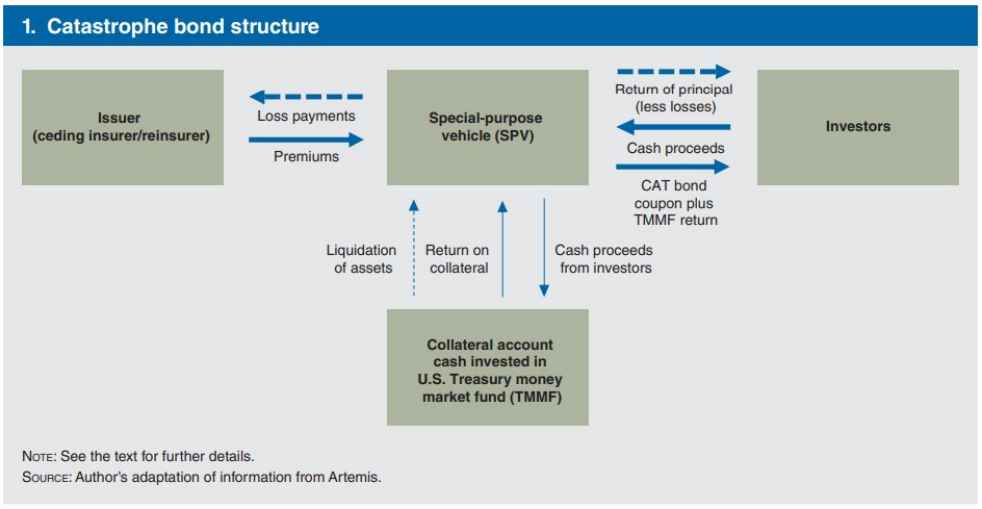


Figure 1: CAT bond structure (Polack).

won't be as affected by other financial markets. By investing in CAT bonds, the investors can diversify their portfolio. And if no catastrophic event takes place, their principle will be returned, including the coupons (Polack).

2.2 Barrier options

In this thesis, we will focus on the parametric trigger. This trigger will be payed out after it surpasses a threshold. To mimic this threshold we can use barrier options.

Definition 2.1. *An **option** is a contract written by a seller, that gives the right (but not the obligation) to the holder to trade in the future the underlying asset at the a previously agreed price (Oosterlee, 2022, slide 12).*

An European **call option** is the option of the buyer to buy a certain asset, S, for a certain strike price, K. The writer of the contract must sell the certain asset if the buyer chooses to buy the asset. The European **put option** is the option for the buyer to sell the asset at a certain predetermined time for a predetermined strike price. And the seller of the option must then buy the asset when the buyer chooses to sell (Oosterlee, slide 12). The predetermined time for an option to end is also called the expiry date. The value of the European call option at expiry date T is given by:

$$V_c(T) = \max(S(T) - K, 0). \tag{1}$$

The value of the European put option at expiry date T is given by:

$$V_p(T) = \max(K - S(T), 0), \tag{2}$$

(Oosterlee, slide 14). A barrier option is an exotic option where the payoff is determined by a fixed barrier. There are four types of barrier options. Knock-in, divided into two classes, up-and-in and down-and-in. A knock-in comes to life when the barrier is past

during the lifespan of the option. Once the barrier is past, the option stays into existence until the option expires. For the **up-and-in** the barrier is above the starting point of the asset. For the **down-and-in**, the barrier is below.
The knock-out is also divided into two classes, up-and-out and down-and-out. If the derivative surpasses the barrier in the knock-out, it ceases to exists. For the **up-and-out** the barrier is above the starting point, for the **down-and-out** below (Chen, 2022).
All barrier options can be either a call or a put options, we will examine the up-and-out barrier call option. The payoff for an up-and-out barrier call option, for a barrier B, an asset S, strike price K and expiry date T is then given by:

$$\begin{aligned} V_c^{uo}(T) &= \max(S(T)-K,0)\mathbb{1}_{\{\max\limits_{0\le t\le T} S(t)<B\}} \\ &= \begin{cases} \max(S(T)-K,0), & \text{if } \max\limits_{0\le t\le T} \quad S(t) < B \\ 0, & \text{if } \max\limits_{0\le t\le T} \quad S(t) \ge B. \end{cases} \end{aligned} \tag{3}$$

We can use this for every type of barrier option, by only changing the if-statements. For put options we would need to change the $\max(S(T)-K,0)$ to $\max(K-S(T),0)$ (Privault, 2024).

2.3 Valuating barrier options

In equation (3), we have seen how to define the payoff of a barrier option. In this section we will further expand on this and valuate the barrier options at a time t, for $0 \le t \le T$, where T is the expiry date. This part is inspired by the book Stochastic Calculus for Finance 2, by Steven E. Shreve (Shreve, 2004), section 7.3.

Before we continue, we need to specify a few definitions that are needed from now on. One of them is a probability triple, $(\Omega, \mathcal{F}, \mathbb{P})$, for which $\mathcal{F}$ is a σ-algebra, $\mathbb{P}$ the probability measure and X a random variable.

Definition 2.2. *A **probability triple** $(\Omega, \mathcal{F}, \mathbb{P})$ is defined by Ω as the sample space, $\mathcal{F}$ as a measurable set and $\mathbb{P}$ as a probability measure (Ruszel, Spitoni, 2024. Definition 1.1).*

Definition 2.3. *$\mathcal{F}$ is called a σ-algebra, if $\mathcal{F} \subset \{A | A \subset \Omega\} = 2^\Omega$ and if:*
(i) $\emptyset, \Omega \in \mathcal{F}$,
(ii) If $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$,
(iii) For a countable set $I = \{1, 2, ..\}$ and $A_1, ..., A_i, ... \in \mathcal{F}$, then $\cup_{i\in I} A_i \in \mathcal{F}$ (Ruszel, Spitoni. Definition 1.2).

Definition 2.4. *$\mathbb{P}$ is a **probability measure** if:*
(i) $\mathbb{P}$ is a map from $\mathcal{F}$ to $[0, 1]$,
(ii) $\mathbb{P}(\Omega) = 1$,
(iii) Let $\{A_n\}_{n\ge1}$ be a disjoint collection, then

$$\mathbb{P}\left(\cup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n),$$

i.e. $\mathbb{P}$ *is countably additive (Ruszel, Spitoni. Definition 1.3).*

Definition 2.5. *A* ***random variable*** X *is a measurable function from the sample space to* $\mathbb{R}$*, i.e.*

$$X : (\Omega, \mathcal{F}) \to (\mathbb{R}, \mathcal{B}),$$

where $\mathcal{B}$ *is the Borel* σ*-algebra of* $\mathbb{R}$ *generated by the open sets of* $\mathbb{R}$*. So we have*

$$X^{-1}(B) = \{\omega \in \Omega | X(\omega) \in B\} \in \mathcal{F}$$

(Ruszel, Spitoni. Definition 1.10).

Let's further examine the payoff of a up-and-out call option. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let T be the expiry date, and let $\mathcal{F}(t)$, $0 \leq t \leq T$, be a filtration of a σ-algebra $\mathcal{F}$.

Definition 2.6. *For a non-empty sample space* Ω*. Assume that for* $t \in [0, T]$*, where* T *is fixed,* $\mathcal{F}(t)$ *is a* σ*-algebra. If* $s \leq t$*, assume that for every set in* $\mathcal{F}(s)$ *it is also in* $\mathcal{F}(t)$*. We then define the collection of* σ*-algebras* $\mathcal{F}(t)$*, for* $t \in [0, T]$*, a* ***filtration*** *(Shreve. Definition 2.1.1).*

To define the payoff function of the up-and-out barrier call option the paragraph 7.3.1 of the book of Shreve is used. Let $S(t)$ be the stock price, then $S(t)$ depends on the rate interest r and the volatility σ.

Definition 2.7. *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a probability space. Assume that for each* $\omega \in \Omega$*, there exists a continuous function* $W(t)$*, for* $t \geq 0$*, where* $W(0) = 0$ *and that depends on* ω*. Then* $W(t)$*, for* $t \geq 0$*, is a* ***Brownian motion*** *if, for all* $0 = t_1 < t_2 < ... < t_m$*, the increments*

$$W(t_1) = W(t_1) - W(0), W(t_2) - W(t_1), ..., W(t_m) - W(t_{m-1})$$

are independent and for each of these increments holds that they are normally distributed with

$$\mathbb{E}[W(t_{i+1} - W(t_i)] = 0,$$
$$Var[W(t_{i+1} - W(t_i)] = t_{i+1} - t_i,$$

(Shreve. Definition 3.3.1)

Our risky asset follows a geometric Brownian motion, where for $0 \leq t \leq T$ and $\widetilde{W}(t)$ is a Brownian motion under the risk-neutral measure $\widetilde{\mathbb{P}}$,

$$dS(t) = rS(t)dt + \sigma S(t)d\widetilde{W}(t). \tag{4}$$

Then, the solution for the stochastic differential equation for the asset price is

$$S(t) = S(0)e^{\sigma\widehat{W}(t)+(r-\frac{1}{2}\sigma^2)t} = S(0)e^{\sigma\widehat{W}(t)}, \tag{5}$$

with

$$\widehat{W}(t) = (\frac{1}{\sigma}(r - \frac{1}{2}\sigma^2))t + \widetilde{W}(t).$$

We denote $\widehat{M}(T) = \max_{0\leq t\leq T} \widehat{W}(t)$, such that

$$\max_{0\leq t\leq T} S(t) = S(0)e^{\sigma\widehat{M}(T)}.$$

For a knock-out barrier option, we have a payoff if and only if $S(t) \leq B$, or $S(0)e^{\sigma\widehat{M}(T)} \leq B$. So we have a payoff

$$\begin{aligned} V_c^{uo}(T) &= \max(S(0)e^{\sigma\widehat{M}(T)} - K)\mathbb{1}_{\{S(0)e^{\sigma\widehat{M}(T)}\leq B\}} \\ &= (S(0)e^{\sigma\widehat{M}(T)} - K)\mathbb{1}_{\{S(0)e^{\sigma\widehat{W}(t)}\geq K, S(0)e^{\sigma\widehat{M}(T)}\leq B\}} \\ &= (S(0)e^{\sigma\widehat{M}(T)} - K)\mathbb{1}_{\{\widehat{W}(t)\geq k, \widehat{M}(T)\leq b\}}. \end{aligned} \tag{6}$$

Here is $k = \frac{1}{\sigma} log \frac{K}{S(0)}$ and $b = \frac{1}{\sigma} log \frac{B}{S(0)}$.

In the following part section 7.3.2 of the book of Shreve is used. The price for the up-and-out call satisfies a Black-Scholes-Merton partial differential equation that has been changed to account for the barrier, if the asset price follows a Geometric Brownian Motion (GBM). With the partial differential equation, the value of the call should depend on the time and on the value of the asset price at that time. It should also depend on the interest rate r, the volatility σ, the strike price K and the barrier B, however these are constants. The Black-Scholes-Merton partial differential equation for an up-and-out call is explained in Theorem 2.8.

Theorem 2.8. *Let $V_c^{uo}(t,x) = V(t,x)$ refer to the price at time t of the up-and-out call under the assumption that the call has not been knocked out before time t and let $S(t) = x$. Then $V(t,x)$ satisfies the Black-Scholes-Merton partial differential equation*

$$V_t(t,x) + rxV_x(t,x) + \frac{1}{2}\sigma^2 x^2 V_{xx}(t,x) = rV(r,x) \tag{7}$$

where V_i is the price differentiated to i. Then (7) sits in the rectangle $\{(t,x); 0 \leq t \leq T, 0 \leq x \leq B\}$ and has the boundary conditions

$$V(t,0) = 0, \quad 0 \leq t \leq T \tag{8}$$

$$V(t,B) = 0, \quad 0 \leq t \leq T \tag{9}$$

$$V(T,x) = max(x-K,0), \quad 0 \leq x \leq B, \tag{10}$$

(Shreve. Theorem 7.3.1).

Proof theorem 2.8:

Proof. We determine the differential:

$$\begin{aligned} d(e^{-rt}v(t,S(t))) &= e^{-rt}[-rv(t,S(t))dt + v_t(t,S(t))dt + v_x(t,S(t))dS(t) \\ &+ \frac{1}{2}v_{xx}(t,S(t))dS(t)dS(t)] \\ &= e^{-rt}[-rv(t,S(t)) + v_t(t,S(t)) + rS(t)v_x(t,S(t)) \\ &+ \frac{1}{2}\sigma^2 S^2(t)v_{xx}(t,S(t))]dt \\ &+ e^{-rt}\sigma S(t)v_x(t,S(t))d\widetilde{W}(t). \end{aligned} \tag{11}$$

Before the option knocks out, the dt term must be zero for $0 \leq t \leq \rho \leq T$. However since (t, S(t)) can reach any point in $\{(t,x); 0 \leq t \leq T, 0 \leq x \leq B\}$ before the option knocks out, the Black-Scholes-Merton equation (7) must hold for every $t \in [0,T]$ and $x \in [0,B]$ (Shreve. Proof of theorem 7.3.1). □

The first boundary (8), says that if the asset price is zero, so is the payoff. The second boundary (9) describes that when the boundary is surpassed, the payoff will be zero. So, the Black-Scholes-Merton equation is relevant unless the boundary is crossed. If the boundary is not crossed, then by (10), we have the same payoff as equation (6).
In all cases of the up/down-and-in/out call and put options, an analytical formula for the option value can be obtained by solving the Black-Scholes-Merton partial differential equation with the appropriate final time T and the boundary conditions. However, we will still be looking at the up-and-out call. The analytical solution of the payoff of the up-and-out call barrier option is retrieved for section 7.3.3 from the book of Shreve.
Let $t \in [0,T)$, $\tau = T - t$ and assume that the underlying asset price at time t is $S(T) = x$, where $0 < x \leq B$. Then, the analytical call price as a function of $V_c^{ou}(t,x)$ is:

$$\begin{aligned} V_c^{ou}(t,x) &= x[N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{x}{K}\right) + (r + \frac{1}{2}\sigma^2)\tau\right]\right) - N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{x}{B}\right) + (r + \frac{1}{2}\sigma^2)\tau\right]\right)] \\ &- e^{-r\tau}K[N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{x}{K}\right) + (r - \frac{1}{2}\sigma^2)\tau\right]\right) - N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{x}{B}\right) + (r - \frac{1}{2}\sigma^2)\tau\right]\right)] \\ &- B\left(\frac{x}{B}\right)^{-\frac{2r}{\sigma^2}}[N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{B^2}{Kx}\right) + (r + \frac{1}{2}\sigma^2)\tau\right]\right) \\ &- N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{B}{x}\right) + (r + \frac{1}{2}\sigma^2)\tau\right]\right)] \\ &+ e^{-r\tau}K\left(\frac{x}{B}\right)^{-\frac{2r}{\sigma^2}+1}[N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{B^2}{Kx}\right) + (r - \frac{1}{2}\sigma^2)\tau\right]\right) \\ &- N\left(\frac{1}{\sigma\sqrt{\tau}}\left[log\left(\frac{B}{x}\right) + (r - \frac{1}{2}\sigma^2)\tau\right]\right)]. \end{aligned} \tag{12}$$

Equation (12) is derived from equation 7.3.20 in Stochastic Calculus for Finance 2, by Shreve. In equation (12), we use the standard cumulative normal distribution property $N(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{\xi^2}{2}} d\xi$.

Equation (12) does satisfy all boundary conditions of Theorem 2.8. When $x > B$ for $t \in [0, T]$, we have $V_c^{ou}(t, x) = V(t, x) = 0$, because the option knocks out when the asset price exceeds the barrier. When the asset price reaches the barrier almost surely, $V_c^{ou}(t, B) = V(t, B) = 0$ because GMB starting at 0 stays at zero and hence the call expires out of the money. If the option doesn't knock out prior to the expiration date, then the payoff will be $V_c^{ou}(T, x) = \max(S(T) - K, 0)$.

3 The Weibull distribution

Traditional insurance cannot cover the economic losses caused by a natural disaster, like an earthquake. According to E. A. A. Ismail: "Globally, a wide gap exists between the economic losses caused by earthquakes, and the losses covered by insurance" (Ismail, 2016). Hence, for insurance companies it is very important to invest in Catastrophic bonds to cover this gap. This problem is also relevant for the Netherlands, in Groningen earthquakes happen because gas is extracted from the ground. F. P. Pijpers published a paper on the earthquakes in Groningen, where he used the dataset from the earthquake catalogue published by the KNMI. He says that: "It appears that the Weibull distribution is appropriate as a descriptor of the observed distribution functions of time intervals for the several regions, and for the different epochs, and that the parameter of the distribution function are different between different zones and epochs" (Pijpers, 2018).
We are going to look closely to the Weibull distribution, which is the statistical process that is often used for earthquakes, specifically shocks and after shocks (Pijpers, p. 6). Let X be a random variable and let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. The Weibull probability density function (PDF) for scale parameter $\alpha > 0$, shape parameter $\beta > 0$ and $y > 0$ is:

$$f_X(y) = \frac{\beta}{\alpha}\left(\frac{y}{\alpha}\right)^{\beta-1} \exp\left\{-\left(\frac{y}{\alpha}\right)^{\beta}\right\}. \tag{13}$$

Here y could be the time length of the earthquake or, as Pijpers used it, the intervals between events of earthquakes with a magnitude greater or equal to 1.2. The cumulative distribution function of the Weibull distribution is

$$F_X(y) = 1 - \exp\left\{-\left(\frac{y}{\alpha}\right)^{\beta}\right\}, \tag{14}$$

also for $y > 0$ and scale and shape parameter $\alpha, \beta > 0$ (Giertz, n.d. P. 141).

3.1 Properties

For a continuous probability distribution, $f(x)$ the expected value is

$$\mathbb{E}(x) = \int_{-\infty}^{\infty} x f(x) dx,$$

(Rice, 2007. P. 118). We find the expected value of the Weibull distribution, for $y > 0$,

$$\mathbb{E}(y) = \int_0^{\infty} y \frac{\beta}{\alpha}\left(\frac{y}{\alpha}\right)^{\beta-1} \exp\left\{-\left(\frac{y}{\alpha}\right)^{\beta}\right\} = \alpha\Gamma\left(1 + \frac{1}{\beta}\right), \tag{15}$$

with the Gamma function (18) (Pijpers). The characteristic function of the Weibull distribution is given by

$$\varphi(ty) = \mathbb{E}(\exp(ity)) = \int_{-\infty}^{\infty} \exp(ity) f_X(y) dy \tag{16}$$

where $i = \sqrt{-1}$ and $t \in \mathbb{R}$ (Rice. P. 161). What is interesting about the characteristic function is in probability the function corresponds to the Fourier transform in analysis (Ruszel, Spitoni. P. 66). The characteristic function exists for all distributions because $|e^{ity}| < 1$ for all values of t (Rice. P. 161).
For the Weibull distribution the characteristic function is:

$$\varphi(ty) = \frac{\beta}{\alpha} \int_0^{\infty} \left[\cos(ty) \left(\frac{y}{\alpha}\right)^{\beta-1} \exp\left\{ -\left(\frac{y}{\alpha}\right)^{\beta} \right\} + i \sin(ty) \left(\frac{y}{\alpha}\right)^{\beta-1} \exp\left\{ -\left(\frac{y}{\alpha}\right)^{\beta} \right\} \right] dy.$$

And so we have found,

$$\varphi(ty) = 1 + \sum_{n=0}^{\infty} \frac{(it\alpha)^{n+1}}{n!\beta} \Gamma\left(\frac{n+1}{\beta}\right), \tag{17}$$

where $\Gamma(a)$ is the Gamma function given by

$$\Gamma(a) = \int_0^{\infty} u^{a-1} e^{-u} du, \tag{18}$$

(Muraleedharan et al, 2007).

3.2 Maximum likelihood estimator

Pijpers uses the maximum likelihood estimator (MLE) to estimate the parameters α and β. The MLE will be further explained in this section, after which we will apply this to the Weibull distribution.
We can find the scale and shape parameter of the PDF using the MLE. To use this method we need a likelihood function.

Definition 3.1. *Let $\mathbf{X} = (X_1, ..., X_n)$ be a random vector with probability density function (or probability mass function) $f(\mathbf{x}; \theta)$ which depends on a parameter $\theta \in \Theta$. For a fixed $\mathbf{x}$, the function:*

$$\theta \to L(\theta; \mathbf{x}) := f(\mathbf{x}; \theta) \tag{19}$$

*is called **the likelihood function**. For the random vector $\mathbf{X} = (X_1, ..., X_n)$ of independent and identically distributed random variables, the joint likelihood function is*

$$L(\theta; \mathbf{x}) = L(\theta; x_1, ..., x_n) = \prod_{i=1}^{n} f(x_i; \theta), \tag{20}$$

(Ruszel, Spitoni. Definition 2.20).

We can estimate θ by finding the maximum of the likelihood function. This brings us to the following definition:

Definition 3.2. *Let $\mathbf{X} = (X_1, ..., X_n)$ be a random sample defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P}_\theta)$. The maximum likelihood estimate $S_n(\mathbf{x}) \in \Theta \subset \mathbb{R}^d$ is the value that maximises the likelihood function $L(\theta; \mathbf{x})$ for $\mathbf{x} = (x_1, ..., x_n)$. The MLE is defined as $S_n(\mathbf{X})$:*

$$S_n(\mathbf{X}) := argmax_{\theta \in \Theta}\{L(\theta; \mathbf{X})\},$$

(Ruszel, Spitoni. Definition 2.11).

To find the MLE we need to find the extreme points. We can take the logarithm of the likelihood function. This works because the logarithm is a monotone function, $\hat{\theta}$ maximises $\theta \to L(\theta; \mathbf{x})$ if and only if it maximises $\theta \to \log(L(\theta; \mathbf{x}))$. If $L(\theta; \mathbf{x})$ is differentiable in $\theta \in \Theta \subset \mathbb{R}^d$ and takes its maximum in an interior point $\hat{\theta}$ of Θ, then

$$\frac{\partial}{\partial \theta_j} \log(L(\theta; \mathbf{x}))\,|_{\theta=\hat{\theta}} = 0$$

for $j = 1, .., d$ (Spitoni, 2024. Slide 4).
In paragraph 2.4 of Pijpers' paper he gives us the log-likelihood function and the MLE for α and β, which we will explain in this section. For the Weibull distribution we get the likelihood function

$$L(\alpha, \beta; \mathbf{y}) = \prod_{i=1}^{n} f(y_i; \alpha, \beta) = \prod_{i=1}^{n} \frac{\beta}{\alpha} \left(\frac{y_i}{\alpha}\right)^{\beta-1} \exp\left\{-\left(\frac{y_i}{\alpha}\right)^{\beta}\right\}, \tag{21}$$

for $y_i, \alpha, \beta > 0$ for all $i = 1, ..., n$. We then find the log-likelihood

$$\log(L(\alpha, \beta; \mathbf{y})) = n\left[\log\left(\frac{\beta}{\alpha}\right) - (\beta - 1)\log(\alpha)\right] + (\beta - 1)\sum_{i=1}^{n} \log(y_i) - \sum_{i=1}^{n} \left(\frac{y_i}{\alpha}\right)^{\beta}. \tag{22}$$

We can now solve this by differentiating to the parameters α and β and so we find the derivatives:

$$\frac{\partial}{\partial \alpha} \log(L(\alpha, \beta; \mathbf{y})) = \frac{\beta}{\alpha}\left[-n + \sum_{i=1}^{n} \left(\frac{y_i}{\alpha}\right)^{\beta}\right] \tag{23}$$

$$\frac{\partial}{\partial \beta} \log(L(\alpha, \beta; \mathbf{y})) = \frac{n}{\beta} - n\log(\alpha) - \left(\frac{1}{\alpha}\right)^{\beta} \sum_{i=1}^{n} y_i^{\beta} \log(y_i) + \left(\frac{1}{\alpha}\right)^{\beta} \log(\alpha) \sum_{i=1}^{n} y_i^{\beta} + \sum_{i=1}^{n} \log(y_i). \tag{24}$$

When we set (25) and (26) to zero, substitute (25) into (26) and after rearranging we find

$$\log(\alpha) = \frac{1}{\beta} \log\left[\frac{1}{n}\sum_{i=1}^{n} y_i^{\beta}\right]$$

$$\frac{1}{\beta} = \sum_{i=1}^{n} \left(\frac{y_i^{\beta}}{\sum_{i=1}^{n} y_i^{\beta}} - \frac{1}{n}\right) \log(y_i)$$

The maximum likelihood estimate for β is biased.

Theorem 3.3. *A function $\hat{\theta}$ of the sample $X_1, ..., X_n$ is said to be* ***unbiased*** *if for the population parameter θ the expectation $\mathbb{E}(\hat{\theta}) = \theta$ (Rice. P. 206).*

Pijpers gives us the bias correction

$$\beta_{\text{corr}} = \beta \frac{1}{1 + \frac{1.37}{n-2.92}\sqrt{\frac{n}{n-1}}}. \tag{25}$$

Using the data from the Groningen earthquakes in the MLE the parameter estimates for $\alpha = 118$ and $\beta = 1.425$ are found for the zone South-East of the gas fields (Pijpers, figure 2.1). In Figure 2 the Weibull distribution with these parameters is plotted. This plot shows the time intervals between events of earthquakes with a magnitude $M \geq 1.2$ (Pijpers). The long tail in Figure 2 is caused by the aftershocks of the earthquakes.

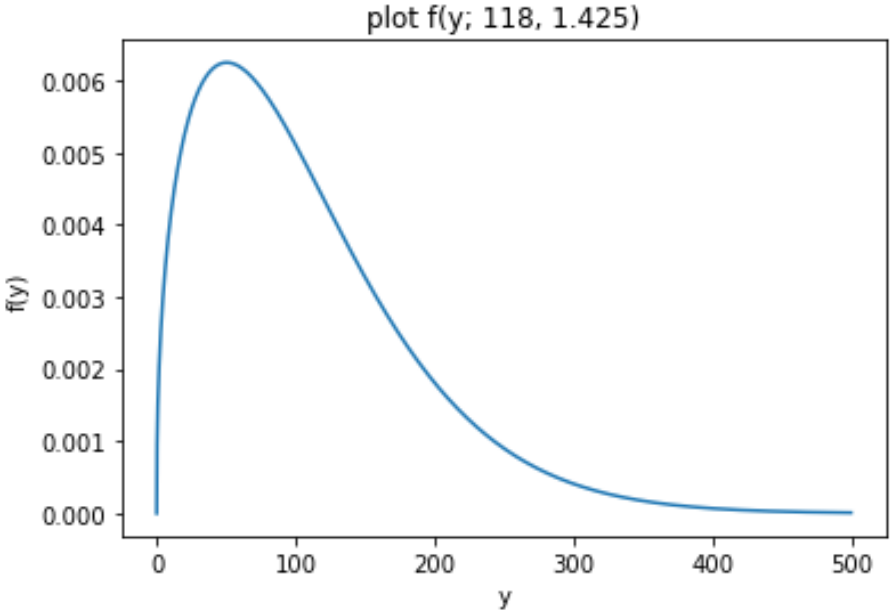


Figure 2: The Weibull distribution for $\alpha = 118$ and $\beta = 1.425$.

4 The COS method

This chapter is inspired by the book Mathematical Modeling and Computation in Finance chapter 6 by C.W. Oosterlee and L. A. Grzelak (Oosterlee, Grzelak, 2019). In this chapter we will look at the COS method applied to European option valuation. After this, we will try and apply this same numerical method to the Weibull distribution.

4.1 The COS method for approximating the probability density function

In this section paragraph 6.1.2 from the book Mathematical Modeling and Computation in Finance by Oosterlee and Grzelak will be discussed.

The Fourier cosine expansion gives a favourable estimation for functions whose domain is finite. The general definition of the Fourier expansion of a function $g(x)$ on an interval $[-1, 1]$ is,

$$g(\theta) = \sum_{k=0}^{\infty}{}' A_k \cos(k\pi\theta) + \sum_{k=0}^{\infty} B_k \sin(k\pi\theta), \tag{26}$$

where the prime on at the first summation implies that the first sum must be multiplied by $\frac{1}{2}$. Here

$$A_k = \int_{-1}^{1} g(\theta)\cos(k\pi\theta)d\theta,$$

$$B_k = \int_{-1}^{1} g(\theta)\sin(k\pi\theta)d\theta.$$

When $B_k = 0$ we have the classical Fourier cosine expansion, for which we can express even functions around $\theta = 0$. The classical Fourier cosine expansion is used in the COS method on the interval $[-\pi, \pi]$, with π as a scaling factor and the functions maximum is attained on this domain. For functions on other finite intervals, such as $[a, b] \in \mathbb{R}$, the Fourier cosine series can be obtained by a change of variables:

$$\theta := \frac{y-a}{b-a}\pi, \quad y = \frac{b-a}{\pi}\theta + a.$$

Any real finite supported function can be estimated by the cosine expansion. A PDF where X is a random variable with probability space $(\Omega, \mathcal{F}, \mathbb{P})$,

$$f_X(y) = \frac{1}{2\pi}\int_{\mathbb{R}} e^{-iuy}\varphi_X(u)du, \tag{27}$$

is such a real function with finite support. This is because for $y \to \pm\infty$, $f_X(y) \to 0$. Because of the finite support we can alter the boundaries of the integral. We use the characteristic function of the PDF we want to compare. To approximate the PDF we will apply the Fourier cosine series on the characteristic function of the PDF. The characteristic function of the PDF is

$$\varphi_X(u) = \int_{\mathbb{R}} e^{iuy} f_X(y)dy. \tag{28}$$

By altering the boundaries of the integral, we get the characteristics function

$$\hat{\varphi}_X(u) = \int_a^b e^{iuy} f_X(y)dy \approx \int_{\mathbb{R}} e^{iuy} f_X(y)dy = \varphi_X(u). \tag{29}$$

Recall the Euler formula $e^{iu} = \cos(u) + i\sin(u)$, by taking the real part (denoted $Re\{.\}$) of the Euler formula we get $\cos(u)$. Let X be any random variable and $a \in \mathbb{R}$ be a constant. Then

$$\varphi_X(u)e^{ia} = \int_{\mathbb{R}} e^{i(uy+a)} f_X(y)dy. \tag{30}$$

If take the real part of (30), we are close to the Fourier expansion of the characteristic function. When we substitute $u = \frac{k\pi}{b-a}$, for $k \in \mathbb{N}$, multiply (29) by $e^{-i\frac{ka\pi}{b-a}}$ and then take the real part of this we get:

$$Re\left\{\hat{\varphi}_X\left(\frac{k\pi}{b-a}\right)e^{-i\frac{ka\pi}{b-a}}\right\} = \int_a^b \cos\left(k\pi\frac{y-a}{b-a}\right)f_X(y)dy. \tag{31}$$

At the right-hand side of (31) we have found

$$A_k \equiv \frac{2}{b-a}Re\left\{\hat{\varphi}_X\left(\frac{k\pi}{b-a}\right) * \exp\left(-i\frac{ka\pi}{b-a}\right)\right\}. \tag{32}$$

Using (29) we can rewrite (32) and find:

$$F_k \equiv \frac{2}{b-a}Re\left\{\varphi_X\left(\frac{k\pi}{b-a}\right) * \exp\left(-i\frac{ka\pi}{b-a}\right)\right\}. \tag{33}$$

And so, we have found the numerical approach of the PDF on $[a, b]$:

$$\hat{f}_X(y) \approx \sum_{k=0}^{\infty}{}'F_k \cos\left(k\pi\frac{y-a}{b-a}\right) \approx \sum_{k=0}^{N-1}{}'F_k \cos\left(k\pi\frac{y-a}{b-a}\right). \tag{34}$$

We will approximate the standard normal probability function $f_{N(0,1)}(y) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2}$ by calculating the error with (34). The characteristic function of the standard normal density function is $\varphi_{N(0,1)}(u) = e^{-\frac{1}{2}u^2}$. We calculate the error for different N's in Table 1 and we can see that for a higher N the error becomes significantly small. The values in Table 1 are rounded to 2 decimals. When we plot the characteristic function for different N's and

N	4	8	16	32	64
Error	0.25	0.11	0.01	4.04e-07	3.89e-16

Table 1: The error between the standard normal distribution and the COS method applied to the standard normal distribution.

the normal density function in one image we get Figure 3. We see that for $N = 64$, it seems to align with $f(y)$. We can conclude that for a large N, the COS method is a good estimator for the density function.

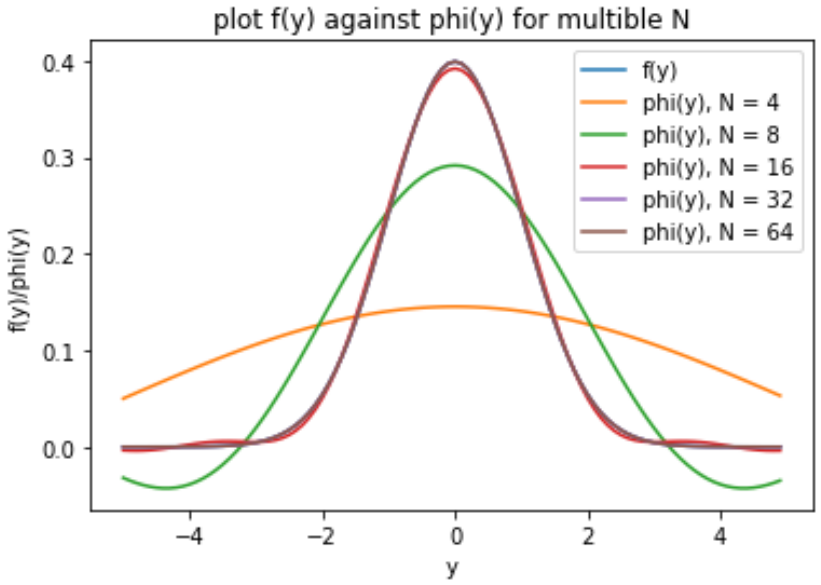


Figure 3: The plot for the standard normal distribution and the COS method applied to the standard normal distribution.

4.2 The COS method for the payoff of an European call option

In a similar way as in section 4.1, we can approximate the pricing European option using the Fourier cosine expansion. In this section paragraph 6.2 from the book of Oosterlee and Grzelak will be discussed.
Let X be a random variable with probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Let $t \in [0, T]$, where T is the expiry date and S is an asset. We set $X(t) = \log(S(t))$, with process $X(t)$ taking values $X(t) = x$ and $X(T) = y$, $\tau = T - t$. Then the value of a plain vanilla option is

$$v_1(t_0, x) = e^{-r\tau}\mathbb{E}[v(T, y)|\mathcal{F}(t_0)] = e^{-r\tau}\int_{\mathbb{R}} v(T, y) f_X(T, y; t_0, x)dy, \tag{35}$$

where $f_X(T, y; t_0, x) = f(y|x)$ is the transition probability density of $X(T)$ and r is the interest rate.
Because the density decays to zero when $y \to \pm\infty$, we can change the boundaries of the integral in (35). The integral will not go over the whole $\mathbb{R}$ but from a to b, $[a, b] \subset \mathbb{R}$. Secondly, we can replace the PDF $f(y|x)$ by its cosine expansion in y, then

$$\hat{f}(y|x) = \sum_{k=0}^{\infty}{}' \overline{A}_k(x) \cos\left(k\pi\frac{y-a}{b-a}\right), \tag{36}$$

with,

$$\overline{A}_k(x) = \frac{2}{b-a}\int_a^b \hat{f}(y|x) \cos\left(k\pi\frac{y-a}{b-a}\right) dy. \tag{37}$$

And so we have

$$v_1(t_0, x) = e^{-r\tau}\int_a^b v(y, T) \sum_{k=0}^{\infty}{}' \overline{A}_k(x) \cos\left(k\pi\frac{y-a}{b-a}\right) dy. \tag{38}$$

We can interchange the summation and the integral, then

$$v_1(t_0,x) = \frac{b-a}{2} e^{-r\tau} \sum_{k=0}^{\infty}{}' \overline{A}_k(x) V_k, \tag{39}$$

for

$$V_k := \frac{2}{b-a} \int_a^b v(T,y) \cos\left(k\pi \frac{y-a}{b-a}\right) dy. \tag{40}$$

We have now found the Fourier cosine series coefficients of the two real function $f(y|x)$ and $v(y,T)$. Since the coefficients decay rapidly, we can change the upper boundary of the summation from ∞ to $N-1$ for some $N \in \mathbb{N}$. We find the COS pricing formula for the value of the European call option,

$$V(t_0,x) \approx K e^{-r\tau} Re \left\{ \sum_{k=0}^{N-1}{}' \varphi_X\left(\frac{k\pi}{b-a}, T\right) U_k \exp\left(ik\pi \frac{x-a}{b-a}\right) \right\}. \tag{41}$$

Where we have the strike price K, the characterisation function $\varphi_X(y)$ and $x = log(\frac{S_0}{K})$. The analytical cosine series coefficients, χ_k, of $g(y) = e^y$ on the interval $[c,d] \subset [a,b]$,

$$\begin{aligned} \chi_k(c,d) = \frac{1}{1+(\frac{k\pi}{b-a})^2} \Bigg[&\cos\left(k\pi \frac{d-a}{b-a}\right) e^d - \cos\left(k\pi \frac{c-a}{b-a}\right) e^d \\ &+ \frac{k\pi}{b-a} \sin\left(k\pi \frac{d-a}{b-a}\right) e^d - \frac{k\pi}{b-a} \sin\left(k\pi \frac{c-a}{b-a}\right) e^c \Bigg]. \end{aligned} \tag{42}$$

The analytical cosine series coefficient, ψ_k of $g(y) = 1$ on the interval $[c,d] \subset [a,b]$ is,

$$\psi_k(c,d) = \begin{cases} \left[\sin\left(k\pi \frac{d-a}{b-a}\right) - \sin\left(k\pi \frac{c-a}{b-a}\right)\right] \frac{b-a}{k\pi}, & k \neq 0 \\ d-c, & k = 0. \end{cases} \tag{43}$$

With these function we find

$$U_k = \frac{2}{b-a}(\chi(0,b) - \psi(0,b)). \tag{44}$$

As a simple iteration range it is suggested that it is easier to compute that

$$[a,b] = [-L\sqrt{T}, L\sqrt{T}].$$

We will give an example by computing the error for the European call under Geometric Brownian Motion using the COS method. The characteristic function of the Geometric Brownian Motion is $\varphi_X(u,t) = \exp(iu\mu t - \frac{1}{2}\sigma^2 u^2 t)$. Here we have $\mu = r - \frac{1}{2}\sigma^2 - q$. We use the following parameters, $S_0 = 100$, $r = 0.1$, $q = 0$, $T = 0.1$ and $\sigma = 0.25$. In the book Mathematical Modeling and Computation in Finance chapter 6 by C.W. Oosterlee and L. A. Grzelak, a numerical experiment is done to check the relation between the user-defined tolerance level and the width of the interval $[a,b] = [-L\sqrt{T}, L\sqrt{T}]$. It is found that for the Geometric Brownian motion, $L = 8$ is a good fit (Oosterlee, Grzelak. P. 173.). The

N	16	32	64	128	256
K = 80	2.1664	0.10327	0.0013	4.9929e-10	3.2649e-10
K = 100	0.8138	0.0604	0.0003	3.3480e-10	3.2563e-10
K = 120	3.6033	0.2132	0.0020	1.8432e-10	7.2331e-11

Table 2: The error between the European call under GBM and their reference values, for different strike prices $K = 80, 100, 120$, relative 20.799226309, 3.659968453, and 0.044577814.

choice is made to make $t_0 = 0$.
We compute the error for three different strike prices $K = 80, 100, 120$. We will calculate the absolute error between the reference values: 20.799226309, 3.659968453, and 0.044577814 and the found values. The results are presented in Table 2.

The absolute errors in Table 2 are rounded to 4 decimals. We find that the larger the N, the smaller the absolute error is and we can conclude that for a large N the COS method is a good approximator for the European call under the Geometric Brownian motion.

4.3 The COS method for the Weibull distribution

We apply the COS method to the Weibull PDF, equation (13). We use the characteristic function of the Weibull distribution, equation (17), in the COS method, (34). We use the parameter $\alpha = 1$ and $\beta = 5$. To calculate the infinite sum of (17) and the Gamma function, we use the package mpmath in Python. According to Balasubramanian, who looked at different software support for programming languages such as mpmath for Python, mpmath allows the user to preform arithmetic operation with large numbers without losing precision (Balasubramanian, 2023) .
We calculate the error the same way as we did for Table 1.

N	4	8	16	32	64
Error	0.56	0.05	8.65e-5	8.65e-11	1.92e-12

Table 3: The error between the Weibull distribution and the COS method applied to the Weibull distribution.

We find that for a larger N we find that the error becomes really small. The error are rounded to 2 decimals. When we plot the Weibull PDF in the same figure, Figure 4, as the COS method with the iteration $N = 4, 8, 16, 32, 64$, we see that for $N = 64$ the Weibull density function $f(y)$ and $\varphi(ty)$ are overlapping. We can conclude that the COS method is a good approximator for the Weibull density function.

Notice that in Figure 4 the tail is not as heavy. To get a heavier tail, we change the shape parameter β. We plot the Weibull distribution against its COS method for the same N's as in Figure 4. We see that the smaller β is, the heavier the tail. The larger β, the more it looks like the normal distribution.

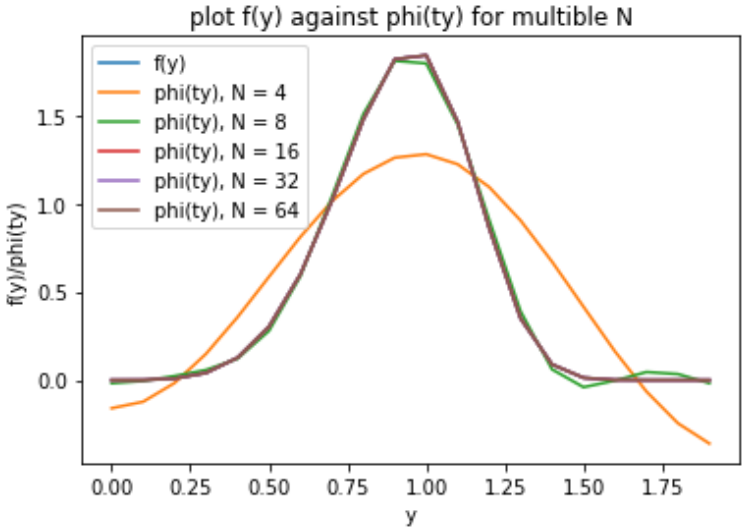
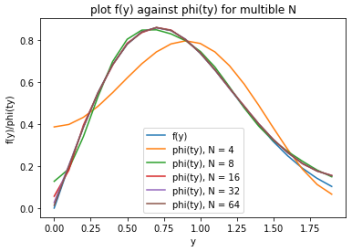
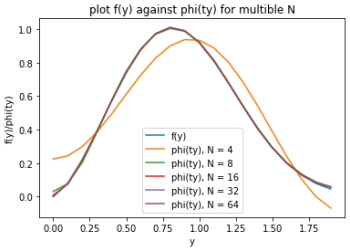


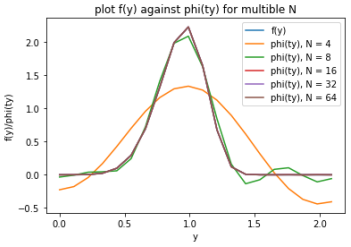
Figure 4: The Weibull distribution for $\alpha = 1$ and $\beta = 5$, together with the COS method of the Weibull distribution for different N's.



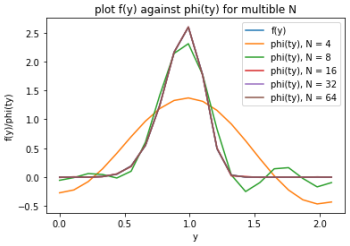
(a) $\beta = 2$.



(b) $\beta = 2.5$.



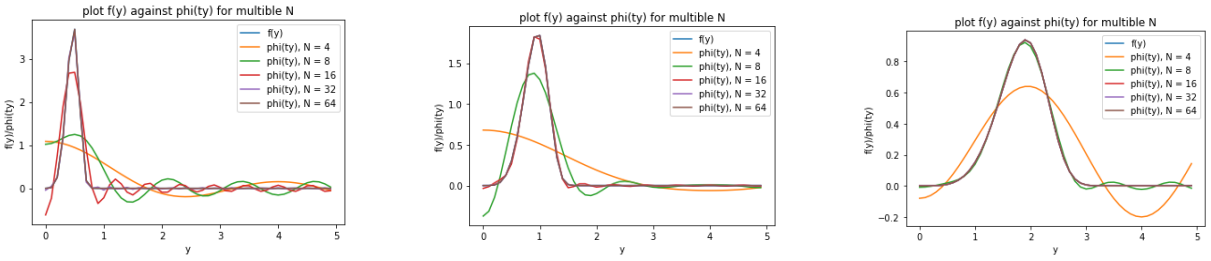
(c) $\beta = 6$.



(d) $\beta = 7$.

Figure 5: The Weibull distribution for different β's

For different α, the scale parameter, we see that the larger α is, the smaller its maximum is. Also the width of the shape changes, for a larger α, the width becomes larger. We also change the intervals of the plotted Weibull distribution. In Figure 4, the Weibull distribution is plotted for $\alpha = 1$ and $\beta = 5$ on the interval $[0, 2]$. This same plot is plotted on the interval $[0, 5]$ in figure 6b. As expected, when changing the interval, nothing changes except for the y-axis length.



(a) $\alpha = 0.5$. (b) $\alpha = 1$. (c) $\alpha = 2$.

Figure 6: The Weibull distribution for different α's

The COS method is also applied to the Weibull distribution with parameters $\alpha = 118$ and $\beta = 1.425$ as found in Pijpers' paper about the Groningen earthquakes. When we plot this together with the COS method for different N's we get figure 7. For Figure 7

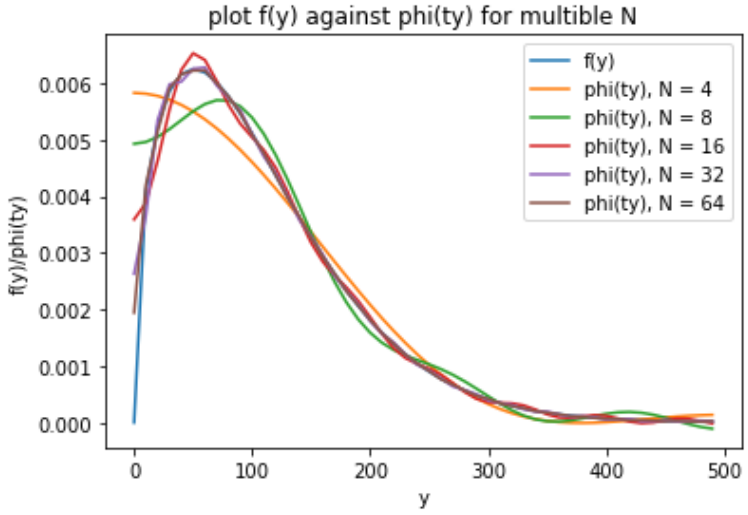


Figure 7: The Weibull distribution for $\alpha = 118$ and $\beta = 1.425$, together with the COS method of the Weibull distribution for different N's.

we will calculate the two-norm. This gives us a better representation of the magnitude of the errors, compared to the absolute error in Table 3. This gives us table 4. Notice in Table 4 that the two-norm decreases when we increase N. So for a larger N, the smaller the error, calculated by using the two-norm, becomes.

N	4	8	16	32	64
Two-norm	0.0064	0.0053	0.0037	0.0027	0.0019

Table 4: The two-norm of the Weibull distribution for $\alpha = 118$ and $\beta = 1.425$, for different N's.

5 Valuing barrier option with the COS method

In this section we will valuate the barrier option using the COS method. We will first introduce the theory behind computing discretely-monitored barrier options. This algorithm will be applied to the Geometric Brownian motion and the results will be compared to the results of the analytical solution of the Black-Scholes-Merton, equation (12). After this, we will have all the tools to compute the discretely-monitored barrier option using the Weibull distribution.

5.1 Discretely-monitored barrier option

This section is inspired by the paper Pricing early-exercise and discrete barrier options by Fourier-cosine series expansion, by F. Fang and C. W. Oosterlee, 2009 (Fang, Oosterlee, 2009). All formulas, lemmas and theory behind this are from this article.

The payoff for a discretely-monitored up-and-out barrier option is (see equation (3)),

$$V_{(uo)} = \max(\alpha(S(T) - K), 0)\mathbb{1}_{\{S(t_i)<B\}}, \tag{45}$$

where B is the barrier, K the strike price and $S(T)$ the asset price at expiry date T. For a call option $\alpha = 1$ and for a put option $\alpha = -1$. The up-and-out option is observed M times at M different observation dates $\Pi = \{t_1, ..., t_M\}$, with $t_1 < .... < t_M = T$. The formula of the price of this M times observed up-and-out option is:

$$\begin{cases} c(x, t_{m-1}) = e^{-r(t_m - t_{m-1})} \int_{\mathbb{R}} v(x, t_m) f(y|x) dy, \\ v(x, t_{m-1}) = \begin{cases} 0, & x \geq h, \\ c(x, t_{m-1}), & x < h. \end{cases} \end{cases} \tag{46}$$

Here $h = \ln(B/K)$ and $m = M, M-1, ..., 2$. We say that $c(x, t_{m-1})$ is the continuation value and $v(x, t_{m-1})$ is the option value with $x := \ln(S(t_{m-1})/K)$ and at time t_{m-1}.
The pricing of the barrier option can be achieved in two stages. The first one being the recovery of the Fourier-cosine series of the option value, v, at time t_1. The second applying the COS method for the European option, similar to equation (41), but before the analytical approach of U_k. So we get the formula:

$$v_2(t_0, x) = e^{-r\tau} Re \left\{ \sum_{k=0}^{N-1}{}' \varphi_X \left(\frac{k\pi}{b-a}, T \right) \exp\left(ik\pi \frac{x-a}{b-a} \right) \right\} V_k(t_1). \tag{47}$$

We will find an approximation for $V_k(t_m)$. We have the following lemma:

Lemma 5.1. *(Backward induction for discrete barrier option) Using backward recursion we can find the following numerical approximation for discretely monitored barrier option. For $m = M-1, M-2, ..., 1$,*

$$\hat{V}_k(t_m) = \hat{C}_k(a, h, t_m) \tag{48}$$

with $\hat{C}_k(x_1, x_2, t_m)$ *given by* (67). *If* $h = \ln(B/K) < 0$, *we have*

$$V_k(t_M) = \begin{cases} 0 & \text{for a call,} \\ G_k(a, h) & \text{for a put.} \end{cases} \tag{49}$$

For $h = \ln(B/K) \geq 0$, *we find*

$$V_k(t_M) = \begin{cases} G_k(0, h) & \text{for a call,} \\ G_k(a, 0) & \text{for a put.} \end{cases} \tag{50}$$

There are a few equations from Lemma 5.1 that need more explanation. We start with $\hat{C}_k(a, h, t_m)$, this is the approximation of the equation

$$C_k(X_1, x_2, t_m) := \frac{2}{b-a} \int_{x_1}^{x_2} c(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{51}$$

where $c(x, t_m)$ is the continuation value at time t_m. We can use the COS method for the continuation value, since the continuation of the option is the same as the value of the plain vanilla option (35), see (46). Applying the COS method to this, the same way as we did in section 4.2, we find

$$\hat{c}(x, t_{m-1}) = e^{-r\tau} Re \left\{ \sum_{k=0}^{N-1}{}' \varphi_X\left(\frac{k\pi}{b-a}, T\right) \exp\left(ik\pi \frac{x-a}{b-a}\right) \right\} V_k(t_m), \tag{52}$$

With $V_k(t_m)$ given in lemma 5.1. At time t_M the payoff for the option value is exact. The equation G_k is very similar to V_k in (40), G_k is

$$G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} g(x, t_m) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{53}$$

where $g(x, t_m)$ is the payoff at time t_m. When we look at the option, we look at it at different times between $t \in [0, T]$. The difference between G_k and (40) is that we look at the payoff and not the option value at a certain time.

Theorem 5.2. *The* $G_k(x_1, x_2)$ *in* (53) *can be determined analytically.*

Proof. For $g(x, t_m) \equiv [\pm K(1 - e^x)]^+$, This is a put option, with $x_2 \leq 0$, such that

$$G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} K(1 - e^x) \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{54}$$

and for the call option, with $x_1 \geq 0$,

$$G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} K(e^x - 1) \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{55}$$

For an observation point x_m^*. Where for a put option $x_m^* \leq 0$, and $x_m^* \geq 0$ for call options, for all $t \in [0, T]$. This gives us

$$G_k(x_1, x_2) = \frac{2}{b-a} \alpha K[\chi_k(x_1, x_2) - \psi_k(x_1, x_2)], \tag{56}$$

where $\alpha = 1$ for a call and $\alpha = -1$ for a put, such that

$$\chi_k(x_1, x_2) := \int_{x_1}^{x_2} e^x \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{57}$$

$$\psi_k(x_1, x_2) := \int_{x_1}^{x_2} \cos\left(k\pi \frac{x-a}{b-a}\right) dx. \tag{58}$$

These integral follow the analytical solutions shown in section 4.2, equation (42) and (43). □

Since at time t_M, $V_j(t_M)$ is exact (from equation (49) and (50)), we can use formula (52) and insert $\hat{c}(x, t_{M-1})$ into equation (51). Interchanging summation and integration gives us

$$\hat{C}_k(x_1, x_2, t_{M-1}) = e^{-t\tau} Re \left\{ \sum_{j=0}^{N-1}{}' \varphi_X\left(\frac{j\pi}{b-a}, T\right) V_j(t_M) M_{k,j}(x_1, x_2) \right\}, \tag{59}$$

with the coefficients of $M_{k,j}(x_1, x_2)$ as

$$M_{k,j}(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi \frac{x-a}{b-a}} \cos\left(k\pi \frac{x-a}{b-a}\right) dx, \tag{60}$$

with $i = \sqrt{-1}$, the imaginary number. For $m = M-2, M-3, ..., 1$, we can define

$$\hat{C}_k(x_1, x_2, t_m) := e^{-t\tau} Re \left\{ \sum_{j=0}^{N-1}{}' \varphi_X\left(\frac{j\pi}{b-a}, T\right) \hat{V}_j(t_{m+1}) M_{k,j}(x_1, x_2) \right\}, \tag{61}$$

which is the result of replacing $V_j(t_{m+1})$ in the definition of $C(x_1, x_2, t_m)$ by its numerical approach $\hat{V}_j(t_m)$ as defined as (48). In vector from (48) is

$$\hat{\mathbf{V}}_k(t_m) = \hat{\mathbf{C}}(a, h, t_m), \tag{62}$$

with

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \begin{cases} e^{-r\tau} Re\{M(x_1, x_2)\Lambda\} \mathbf{V}(t_M), & m = M-1, \\ e^{-r\tau} Re\{M(x_1, x_2)\Lambda\} \mathbf{V}(t_{m+1}), & m = 1, 2, ..., M-2, \end{cases} \tag{63}$$

where $\mathbf{V}(t_M)$ is the vector $(V_0(t_M), V_1(t_M), ..., V_{N-1}(t_M))^T$. The matrix-matrix multiplication of $M\Lambda$ is given by the matrix $\{M_{k,j}\}_{k,j=0}^{N-1}$ and the matrix Λ is a diagonal matrix with elements $\{\varphi_X(\frac{j\pi}{b-a})\}_{j=0}^{N-1}$.
We can use the Fast Fourier Transform (FFT) to compute the matrix vector in (63).

Theorem 5.3. $\hat{\mathbf{C}}(x_1, x_2, t_m)$ *as in equation* (63) *can be computed in* $O(N\log_2(N))$ *operations with the help of the Fast Fourier Transform (FFT) algorithm.*

Proof. Remember that we can write $e^{ia} = \cos(a) + i\sin(a)$. Using this we can rewrite (60) and we get the following:

$$M_{k,j}(x_1, x_2) = -\frac{i}{\pi}\left(M^c_{k,j}(x_1, x_2) + M^s_{k,j}(x_1, x_2)\right), \tag{64}$$

where

$$M^c_{k,j}(x_1, x_2) := \begin{cases} \frac{(x_2-x_1)\pi i}{(b-a)}, & k = j = 0, \\ \frac{\exp\left(i(j+k)\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j+k)\frac{(x_1-a)\pi}{b-a}\right)}{j-k} & \text{else}, \end{cases} \tag{65}$$

and

$$M^s_{k,j}(x_1, x_2) := \begin{cases} \frac{(x_2-x_1)\pi i}{(b-a)}, & k = j, \\ \frac{\exp\left(i(j-k)\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(i(j-k)\frac{(x_1-a)\pi}{b-a}\right)}{j-k} & k \neq j. \end{cases} \tag{66}$$

When we insert (64) into (59) and (61), we get a matrix-vector product representation for $\hat{\mathbf{C}}(x_1, x_2, t_m)$,

$$\hat{\mathbf{C}}(x_1, x_2, t_m) = \frac{e^{-r\tau}}{\pi} Im\{(M^c + M^s)\mathbf{u}\}, \tag{67}$$

where $Im\{.\}$ means that we take the imaginary part of the argument, and

$$\mathbf{u} := \{u_j\}_{j=0}^{N-1}, \quad u_j := \varphi\left(\frac{j\pi}{b-a}\right) V_j(t_{m+1}), \quad u_0 = \frac{1}{2}\varphi(0)V_0(t_{m+1}). \tag{68}$$

The matrices

$$M^c := \{M^c_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1} \quad \text{and} \quad M^s := \{M^s_{k,j}(x_1, x_2)\}_{k,j=0}^{N-1},$$

have special structures, so the FFT algorithm can be used for the efficient computation of the matrix-vector products. In particular, the matrix M^c has a special form, a matrix of this form is called a Hankel matrix,

$$M^c = \begin{bmatrix} m_0 & m_1 & m_2 & ... & m_{N-1} \\ m_1 & m_2 & ... & ... & m_N \\ \vdots & & & & \vdots \\ m_{N-2} & m_{N-1} & ... & ... & m_{2N-3} \\ m_{N-1} & ... & ... & m_{2N-3} & m_{2N-2} \end{bmatrix}_{N\times N} \tag{69}$$

and the matrix M^s also has a special form, a matrix of this form is called a Toeplitz matrix,

$$M^s = \begin{bmatrix} m_0 & m_1 & ... & m_{N-2} & m_{N-1} \\ m_{-1} & m_0 & m_1 & ... & m_{N-2} \\ \vdots & & \ddots & & \vdots \\ m_{2-N} & ... & m_{-1} & m_0 & m_1 \\ m_{1-N} & m_{2-N} & ... & m_{-1} & m_0 \end{bmatrix}_{N\times N} \tag{70}$$

with

$$m_j := \begin{cases} \frac{(x_2-x_1)\pi i}{(b-a)}, & j=0 \\ \frac{\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right)-\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)}{j} & j\neq 0. \end{cases} \tag{71}$$

□

The product of the matrix M^s and the vector $\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_s \cdot \mathbf{u}_s$ where the $2N$-vectors are

$$\mathbf{m}_s = [m_0, m_{-1}, m_{-2}, ..., m_{1-N}, 0, m_{N-1}, m_{N-2}, ..., m_1],$$

and

$$\mathbf{u}_s = [u_0, u_1, ..., u_{N-1}, 0, ..., 0]^T.$$

This can be done because of the property of the Toeplitz matrix that it can be transformed into a circular convolution when multiplied by a vector. For the Hankel matrix M^c, there is no such property, hence we have the following result:
The product of the matrix M^c and the vector $\mathbf{u}$ is equal to the first N elements of $\mathbf{m}_c \cdot \mathbf{u}_c$, in reversed order, with the $2N$-vectors:

$$\mathbf{m}_c = [m_{2N-1}, m_{2N-2}, ..., m_1, m_0]$$

and

$$\mathbf{u}_c = [0, ..., 0, u_0, u_1, ..., u_{N-1}].$$

For the efficient computation of $M^c\mathbf{u}$, we need to construct the following circulant matrix, M^u

$$M^u = \begin{bmatrix} 0 & u_{N-1} & u_{N-2} & ... & ... & ... & 0 \\ 0 & 0 & u_{N-1} & u_{N-2} & ... & ... & 0 \\ \vdots & & \ddots & & \ddots & & \vdots \\ 0 & ... & 0 & u_{N-1} & u_{N-2} & ... & u_0 \\ u_0 & 0 & ... & 0 & u_{N-1} & ... & u_1 \\ u_1 & u_0 & 0 & ... & 0 & ... & u_2 \\ \vdots & & \ddots & & \ddots & & \vdots \\ u_{N-2} & .. & u_0 & 0 & ... & 0 & u_{N-1} \\ u_{N-1} & u_{N-2} & ... & u_0 & 0 & ... & 0 \end{bmatrix}_{(2N)\times(2N).} \tag{72}$$

Computation shows that the first N elements of the multiplication of the vectors $\mathbf{u}_c$ and $\mathbf{m}_c$ are equal to $M^c\mathbf{u}$ in reversed order. The circular convolution of two vectors is equal to the inverse discrete Fourier transform of the products of the forward discrete Fourier transforms.
There is a special property of the m_j vectors. We have $m_{-j} = -\overline{m}_j$, where $\overline{m}_j$ is the conjugate of the vector m_j. And for $j \neq 0$ we have:

$$m_{j+N} = \frac{\exp\left(iN\frac{(x_2-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_2-a)\pi}{b-a}\right) - \exp\left(iN\frac{(x_1-a)\pi}{b-a}\right)\cdot\exp\left(ij\frac{(x_1-a)\pi}{b-a}\right)}{j+N}. \tag{73}$$

5.2 The algorithm to calculate barrier options using the COS method

In this section, the algorithm for the discretely-monitored barrier options will be explained. After this, the results will be compared to the analytical call price for the up-and-out barrier option for the Black-Scholes-Merton model for one underlying asset, equation (12).

This algorithm is inspired by the Matlab code of M.J. Ruijter, 2015, where the Black-Scholes up-and-out call is calculated using the COS method (Ruijter, 2015). The code we use is made in Python.
Ruijter uses the Richardson extrapolation. The idea of the Richardson extrapolation is to construct a better approach using the consecutive approaches (van Leeuwen, T. 2022, p. 76). The code starts with initialising the Richardson extrapolation.
Then we initialise the parameters x, h, a, b, and the time steps dt. We then make the array of k, which is an array for of length N, from zero to $N-1$. The array omega which is equal to $\omega = k\frac{\pi}{b-a}$. And we make the array temp, which is omega from 1 to N, so $N-1$ long.
Also the Fourier cosine coefficients for the payoff function, χ and ψ, are initialised.
We need the characteristic function and set the array k from 1 to N, which makes it of length $N-1$.
We make the vectors $\mathbf{m}_s$ and $\mathbf{m}_c$, using the concatenate function from the NumPy package. This function can glue arrays together. Once we have the arrays $\mathbf{m}_s$ and $\mathbf{m}_c$, we apply the fast Fourier transform to them. We use the fft function from the package SciPy. The package "SciPy is a library of numerical routines for the Python programming language that provides fundamental building blocks for modeling and solving scientific problems", according to Virtanen et al. who looked at the capabilities of the Python package (Virtanen et al., 2020). We make another for-loop to recover $\hat{\mathbf{V}}(t_{m-1})$. We compute $\mathbf{u}(t_m)$ using equation (68). Construct $\mathbf{u}_s$ by adding N zeros to $\mathbf{u}(t_m)$. We then apply the inverse fast Fourier transform (IFFT) to the product of the fast Fourier transform (FFT) of $\mathbf{u}_s$ and the fast Fourier transform of $\mathbf{m}_s$, using the ifft function of the package SciPy, and we get $M^s\mathbf{u}$. We only need the first N elements of this array. The same holds for the IFFT that is needed to make the the $M^c\mathbf{u}$. The IFFT is applied to the product of the FFT of $\mathbf{m}_s$, the sgnvector and the FFT of $\mathbf{u}_s$. We again only need the first N elements of this vector and we can use this to make (67).
After this loop we make the option value and apply the four point Richardson extrapolation scheme with $k0 = \frac{1}{2}$, $k1 = 1$ and $k2 = \frac{3}{2}$.

We will now compare the results to the analytical call price of the up-and-out barrier option for different N's and compare the error. We set $S = 100$, $K = 80$, $T = 0.1$, $r = 0.1$ $\sigma = 0.25$ and $B = 120$. We see that the error does get smaller for a larger N, which is a good sign. In Table 5 the results are rounded to 4 decimals, but it is clear that the COS method converges to one solution and faster than the analytical solution. The COS method is already more converged (we only look at the results rounded to 4 decimals) for $N = 256$, while the analytical solutions still differ for $N = 1024$. To program the

N	128	256	512	1024
COS	19.7748	19.7746	19.7746	19.7746
Analytical	19.7926	19.7836	19.7791	19.7768
Error	0.01772	0.0089974	0.004512	0.002259

Table 5: The comparison of the analytical call price of the up-and-out barrier option and the call price for the up-and-out barrier option computed using the COS method.

analytical solution the norm function from SciPy is used.

5.3 Calculating the value of a down-and-in barrier put option for a specific CAT bond

We now know how we can valuate discretely-monitored barrier options, and that we can use these barrier options as the threshold for a parametric trigger. We now want to find the value of a barrier option for a specific CAT bond. For this CAT bond we want to use the information we found in the paper by Pijpers. We use the Weibull distribution to look at the earthquakes in Groningen, specifically in the zone South-East. We want to use the found $\alpha = 118$ and $\beta = 1.425$ that Pijpers found for the zone South-East. We will now want a CAT bond applied to this region and a parametric trigger.

Assume the average house in Groningen (in this South-East region) is €300,000, which is the asset price S. We want to keep this value, so the strike price, K, is €300,000 as well. If the house were to go down to a value of €100,000 (or 150,000, 200,000 and 250,000), then the issuer can collect. So we set the barrier $B = 100,000$, which is our threshold. And we want to find the value of a down-and-in barrier put option.

To make a down-and-in barrier put option, we need to mirror the characteristics function. We do this by multiplying all ty's in the characteristics function (17) by -1. For a barrier put option, by Lemma 5.1, we find that $V_k(t_M) = G_k(a, h)$. We will examine this for different end dates and different interest rates. The parameter $\sigma = 0.25$ and we set $N = 128$.

	T = 5, $r = 0.1$	T = 10, $r = 0.1$	T = 5, $r = 0.01$	T = 10, $r = 0.01$
B = 100	3.79e-19	1.3e-12	2.34e-18	1.66e-11
B = 150	1.99e-08	7.69e-06	8.91e-08	5.49e-05
B = 200	0.002	0.017	0.007	0.079
B = 250	0.88	0.99	1.97	3.32

Table 6: The down-and-in barrier put option for different barriers, end times T and interest rates for the Weibull distribution.

6 Conclusion and discussion

6.1 Conclusion

In chapter 4.3, we saw in Table 4 that for a larger N, the two-norm error between the COS method and the Weibull distribution becomes smaller when N increases. We saw that the COS method is thus a good approximator for the Weibull distribution. In chapter 5.2 we used the code for the up-and-out barrier call option for the Black-Scholes-Merton model for one underlying asset. In Table 5 we can see that this code works as desired, since for an increase in N, the solution converges. Now we know that the COS method gives a good approximation for the Weibull distribution and the code for the COS method for a discretely-monitored barrier option gives a desired result, we can use these to apply this to our made CAT bond.

When we examine Table 6 we can find three conclusions. First, the higher the barrier, B, the higher the value of the down-and-in put option. This can be expected. When we look at the Weibull distribution in Figure 2, we see that this distribution has a very long tail. Since the chance to find data in this tail is very low, and the option value depends on this distribution, there is a very small chance of S to be in this tail. So the lower we put the barrier, the smaller the chance is that is barrier will be hit. And the higher we set the barrier, the larger the chance is that the barrier will be hit. Hence the value is higher for a higher barrier.
Secondly, we see that for a higher T, which is the end time of the bond, the higher the value is. This is also expected, because the longer the duration of the bond, the more chance S has to hit the barrier.
And finally, the lower the interest rate r, the higher the value of the bond is. Since the continuation value $c(x, t_{m-1})$ in equation (47) depends on e^{-r}, for a smaller r, the higher the continuation value would be, and so the higher the option value is.

The value of the bond helps issuers transfer the risk of the extreme event to the financial market. When the value is low, which is in most cases, there is not a lot of risk being transferred. It is unlikely for the barrier to be reached. So for an issuer it is not that interesting to issue this bond. However, on the other end, it is very interesting for investor to invest in a low value down-and-in barrier put option. For investors, if the value is low, the less likely it is for the event to happen. Hence, they are likely to make a profit of this high risk bond. And even if there would be a disaster, if the value of the house doesn't go down to the barrier, the investors could still make a profit.

6.2 Discussion

When programming the code for the characteristic function of the Weibull distribution, I found that the code doesn't work for $\beta < 1$. I wasn't able to fix this problem. This made that the only parameters found by Pijpers I was able to use were for the zone South-East, because he found $\beta > 1$ for this region only.
I found that the code for the discretely-monitored barrier options is very sensitive and

doesn't work very well for large barriers. Depending on what a and b we choose the code can give us a negative solution, even though the value of an option cannot be negative. The code can also give us extremely large values. I wasn't able to find and correct this error. I was only able to change a and b in such a way that the values of the option weren't too extreme low or high.
The value of the down-and-in barrier put option don't change when we change alpha or beta in the code from chapter 5. A change in alpha and beta changes the maximum, width and tail of the distribution. So a change in these parameters would mean a change in option value. However, this doesn't happen and I wasn't able to find this error.
For further research it would be important to understand the code (8.3 and 8.4) much better, so the errors could be fixed.

7 References

Balasubramanian, K. (November, 2023). Software Support for Arbitrary Precision Arithmetic in Programming Languages. Indian Journal of Cryptography and Network Security, Volume 3, Issue 2. https://www.ijcns.latticescipub.com/wp-content/uploads/papers/v3i2/A1425054124.pdf.

Bamieh, B. (April 25, 2022). Discovering transforms: A tutorial on circulant matrices, circular convolution, and the discrete fourier transform. Cornell University. https://arxiv.org/abs/1805.05533. Retrieved on June 15, 2024.

Chen, J. (April 5, 2022). What Is a Barrier Option? Knock-in vs. Knock-out Options. Investopedia. https://www.investopedia.com/terms/b/barrieroption.asp. Retrieved on March 22, 2024.

Delgutte, B., Greenberg, J. (1999). Chapter 4 - The Discrete Fourier Transform. Biomedical Signal and Image Processing. https://web.mit.edu/~gari/teaching/6.555/lectures/ch_DFT.pdf.

Fang, F., Oosterlee, C. W. (May 22, 2009). Pricing early-exercise and discrete barrier options by fourier-cosine series expansion. Numerische Mathematik.

Hayes, A. (August 23, 2022). What is a special purpose vehicle (SPV) and why companies form them. Investopedia. https://www.investopedia.com/terms/s/spv.asp Retrieved on March 21, 2024.

Giertz, F. (n.d.). Analysis and Optimization of a Portfolio of Catastrophe Bonds. Royal Institute of Technology. Master's Thesis. https://www.math.kth.se/matstat/seminarier/reports/M-exjobb14/140609b.pdf. Retrieved on April 24 2024.

Heckbert, P. (January 27, 1998) Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf.

Intergovernmental Panel on Climate Change. (2021). Climate Change 2021: The Physical Science Basis. Contribution of Working Groep 1 to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Summery for Policymakers. https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_SPM.pdf.

Ismail, E. A. A. (June 18, 2016). The Complementary Compound Truncated Poisson-Weibull Distribution for Pricing Catastrophic Bonds for Extreme Earthquakes. British Journal of Economics, Management and Trade.https://journaljemt.com/index.php/JEMT/article/view/569. Retrieved on April 22 2024.

van Leeuwen, T. (2022) Numerieke Wiskunde (WISB251).

Muraleedharan, G., Rao, A. D., Kurup, P. G., Unnikrishnan Nair, N., Mourani Sinha. (June 20, 2007). Modified Weibull distribution for maximum and significant wave height simulation and prediction. Coastal Engineering 54. https://www.sciencedirect.com/science/article/pii/S0378383907000452. Retrieved on May 3 2024.

Niki. (March 6, 2023). Insurance 101: Understanding Catastrophe Bonds (CAT Bonds) — AgentSync. AgentSync. https://agentsync.io/blog/insurance-101/understanding-cat-bonds.

Oosterlee, C. W. (November 17, 2022). Introduction to Financial Mathematics [PowerPoint presentation]. Blackboard. https://uu.blackboard.com/webapps/blackboard

/content/listContent.jsp?course_id=_140563_1&content_id=_4133231_1&mode=reset. Retrieved on June 10, 2024.
Oosterlee, C. W., Grzelak, L. A.(2019) Mathematical Modeling and Computation in Finance.
Papalexiou, S. M. (February 28, 2013). How extreme is extreme? An assessment of daily rainfall distribution tails. Hydrology and Earth System Sciences https://hess.copernicus.org/articles/17/851/2013/hess-17-851-2013.pdf.
Pijpers, F. P. (May, 2018). Weibull fitting of earthquakes in Groningen. Scientific Paper.https://www.cbs.nl/nl-nl/achtergrond/2018/23/weibull-verdelingen-aardbevingen-groningen.
Polack, A. (2018). Catastrophe bonds: A primer and retrospective. Chicago Fed Letter, No. 405. https://www.chicagofed.org/publications/chicago-fed-letter/2018/405#:~:text=a%20catastrophe%20bond.-,A%20CAT%20bond%20is%20a%20security%20that%20pays%20the%20issuer,(on%20the%20Richter%20scale)
Privault, N. (January 10, 2024). Barrier Options. https://personal.ntu.edu.sg/nprivault/MA5182/barrier-options.pdf.
Rice, J. A. (2007). Mathematical Statistics and Data Analysis, third edition. Duxbury advanced series.
Ruijter, M. (2015). Fourier method based on Fourier cosine series and the characteristic function. [Python Code] https://www2.it.uu.se/itwiki.php?page=research/scientific_computing/project/compfin/benchop/original&action=browse
Ruszel, W. M., Spitoni, C. (April 17, 2024). Lecture notes on Mathematical Statistics WISB263. https://uu.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_146786_1&content_id=_4477788_1&mode=reset. Retrieved on June 15, 2024.
Shreve, S. E. (2004). Stochastic Calculus for Finance II: Continuous-Time Models. Springer Science and Business Media.
Spitoni, C. (May 8, 2024). Lecture 5: Consistency and Asymptotic Normality of MLE. [PowerPoint presentation]. Blackboard. https://uu.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_146786_1&content_id=_4477788_1&mode=reset. Retrieved on June 15, 2024.
Vitanen, P., Grommers, R. ..., and SciPy 1.0 Contributors, (March 2020). Scipy 1.0: fundamental algorithms for scientific computing in Python. Nature methods. Volume 17. https://www.nature.com/articles/s41592-019-0686-2. Retrieved on June 16 2024.

8 Code

8.1 Code section 4.1

```
import math
import cmath
import matplotlib.pyplot as plt

y_length = [-5 + i * 0.1 for i in range(int((10)/0.1))]
integration_interval = list(range(-10, 11))
a = integration_interval[0]
b = integration_interval[-1]
ii_length = b-a
S0 = 100
r = 0.1
q = 0
T = 0.1
sigma = 0.25
mu = r - (1/2)*sigma**2 - q

# Make the standard normal density function for a given y
def sndf(y):
    f = (1/math.sqrt(2*math.pi))*math.exp((-1/2)*y**2)
    return f

# Make the COS methode with the truncate series summation
# and the characteristic function

#Make the characteristic funtion:
def char_f(u):
    phi = math.exp((-1/2)*u**2)
    return phi

def Fourier_cosine_expansion(y, N):
    # F_k dak
    f_hat = []
    i = complex(0,1)
    for k in range(0, N-1):
        if k == 0:
            comp_function = char_f(k * math.pi/(ii_length))
            *cmath.exp(-i*(k*a*math.pi)/(ii_length))
            F_hat = (2/(ii_length))*comp_function.real
            f_hat.append((1/2)*F_hat*math.cos(k*math.pi*((y-a)/(ii_length))))
        else:
            comp_function = char_f(k * math.pi/(ii_length))
```

```
            *cmath.exp(-i*(k*a*math.pi)/(ii_length))
            F_hat = (2/(ii_length))*comp_function.real
            f_hat.append(f_hat[k-1] + F_hat*math.cos(k*math.pi
            *((y-a)/(ii_length))))
    return f_hat[-1]

error = []
y1 = []
y2= []
for j in y_length:
    error.append(sndf(j) - Fourier_cosine_expansion(j, 64))
    y1.append(sndf(j))
    y2.append(Fourier_cosine_expansion(j, 64))

print(max(error))

plt.title('plot f(y) against phi(y)')
plt.xlabel('y_length')
plt.ylabel('y-axis')
plt.plot(y_length, y1)
plt.plot(y_length, y2)

plt.show()
```

8.2 Code section 4.2

```
# Definition for the European call under the GBM using the cos method
def V(K, N):
    V = []
    valuelst = []
    a = -8*math.sqrt(T)
    b = 8*math.sqrt(T)
    c = 0
    d = b
    i = complex(0,1)
    x = math.log(S0/K)
    maximum = 0
    for k in range(0, N-1):
        xi = (1/(1+(k*math.pi/(b-a))**2))
            *(math.cos(k*math.pi*((d-a)/(b-a)))*math.exp(d)
            -math.cos(k*math.pi*((c-a)/(b-a)))*math.exp(c)
            + ((k*math.pi)/(b-a))*math.sin(k*math.pi*((d-a)/(b-a)))*math.exp(d)
            -((k*math.pi)/(b-a))*math.sin(k*math.pi*((c-a)/(b-a)))*math.exp(c))
```

```
        if k == 0:
            Phi = (d-c)
        else:
            Phi = (math.sin(k*math.pi*((d-a)/(b-a)))
            -math.sin(k*math.pi*((c-a)/(b-a))))*((b-a)/(k*math.pi))
        Uk = (2/(b-a))*(xi - Phi)
        phi = cmath.exp(i*((k*math.pi)/(b-a))*mu* T
            - (1/2)*sigma**2 * ((k*math.pi)/(b-a))**2 * T)
        if k == 0:
            complex_part = (1/2)*phi*Uk*cmath.exp(i*k*math.pi*((x-a)/(b-a)))
            V.append(complex_part)
        else:
            complex_part = phi*Uk*cmath.exp(i*k*math.pi*((x-a)/(b-a)))
            V.append(V[k-1] + complex_part)
        complex_part_needed = V[-1]
        value = K* math.exp(-r*T)*complex_part_needed.real
        valuelst.append(value)

        #To return the value:
    return value
```

8.3 Code section 4.3

```
# The Gamma function:
def Gamma(a):
    return  gammainc(a, 0, np.inf, regularized=False)

# Pijpers' parameters:
alpha = 118
beta = 1.425

# The Weibull distribution:
def Weibull(x):
    f = (beta/alpha)*((-x/alpha)**(beta-1))*cmath.exp(-(-x/alpha)**beta)
    return f

# The characteristics function of the Weibull distribution:
def char_f(s):
    i = complex(0,1)
    summation = nsum(lambda n: ((i * -s * alpha) ** (n + 1))
                / (factorial(n) * beta) * Gamma((n + 1) / beta), [0, inf])
    phi = 1 + summation
    return phi
```

```
def Fourier_cosine_expansion(y, N):
    # F_k dak
    f_hat = []
    i = complex(0,1)
    for k in range(0, N-1):
        if k == 0:
            comp_function = char_f(k * math.pi/(ii_length))
            *cmath.exp(-i*(k*a*math.pi)/(ii_length))
            F_hat = (2/(ii_length))*comp_function.real
            f_hat.append((1/2)*F_hat*math.cos(k*math.pi*((y-a)/(ii_length))))
        else:
            comp_function = char_f(k * math.pi/(ii_length))
            *cmath.exp(-i*(k*a*math.pi)/(ii_length))
            F_hat = (2/(ii_length))*comp_function.real
            f_hat.append(f_hat[k-1]
                         + F_hat*math.cos(k*math.pi*((y-a)/(ii_length))))
    return sympify(f_hat[-1])

error = []
for j in y_length:
    error.append(Weibull(j).real - Fourier_cosine_expansion(j, 4))

sum_of_squares = sum(x ** 2 for x in error)

# Calculate the square root of the sum of squares
normvalue = math.sqrt(sum_of_squares)
print(normvalue)
```

8.4 Code chapter 5

```
def AnalyticalCallPrice(S, K, T, r, sig, B):
    x = S
    V = []
    for M in 2**np.arange(4, 11):
        dt = T/M
        tau = T - dt
        part = 1/(sig*np.sqrt(tau))
        part1 = (r+ 0.5*sig**2)*tau
        part2 = (r - 0.5*sig**2)*tau

        Vuoc = (x*(norm.cdf(part*((np.log(x/K))+part1))
                - norm.cdf(part*(np.log(x/B)+part1)))
                -np.exp(-r*tau)*K*(norm.cdf(part*(np.log(x/K)+ part2))
                - norm.cdf(part*(np.log(x/B)+part2)))
```

```
                - B*(x/B)**(-2*r/sig**2)
                *(norm.cdf(part*(np.log(B**2/(K*x))+part1))
                - norm.cdf(part*(np.log(B/x)+ part1)))
                + np.exp(-r*tau)*K*(x/B)**(-(2*r/sig**2)+1)
                *(norm.cdf(part*(np.log(B**2/(K*x))+ part2))
                - norm.cdf(part*(np.log(B/x)+ part2))))
            V.append(Vuoc)
        return V

def downinPut(S, K, T, r, sig, B):
    # Loop for Richardson extrapolation
    MRc = 1
    U_Rich = []
    for M in 2**np.arange(4, 8):

        # Parameters
        x = np.log(S/K)
        h = np.log(B/K)

        # Time step
        dt = T/M

        # Interval [a,b]
        a = np.log(60/K)
        b = np.log(140/ K)
        # Number of Fourier cosine coefficients
        N = 2**7
        k = np.arange(N) # N
        omega = k* (np.pi/(b-a))
        temp = omega[1:] #N-1
        # Fourier cosine coeffients payoff function
        omegamina = omega*(a-a) #N-1
        omegahmina = omega*(h-a) #N-1
        sin_omegamina = np.sin(omegamina) #N-1
        sin_omegahmina = np.sin(omegahmina) #N-1
        cos_omegamina = np.cos(omegamina) #N-1
        cos_omegahmina = np.cos(omegahmina) #N-1
        chi = ((cos_omegahmina + omega* sin_omegahmina)*np.exp(h)
               - (cos_omegamina - omega*sin_omegamina)*np.exp(a))
               / (1+ (omega)**2) #N-1
        psi = (sin_omegahmina - sin_omegamina)/omega #N-1
        psi[0] = h-a
        Uk = (chi - psi).real #N-1
        Uk[0] = 0.5 * Uk[0]
```

```
# Characteristic function
cf = char_f(omega) #N

k = np.arange(1, N) #N-1

# Elements m_j
mj = (np.exp(1j*(h-a)*temp)-1)/ k #N-1
mj_0 = 1j *np.pi*(h-a)/(b-a) #1
mj_N = (np.exp(1j*N*np.pi*(h-a)/(b-a))-1)/N #1
mj_minus = -np.conj(mj) #N-1
mfactor1 = np.exp(1j*N*np.pi*(h-a)/(b-a)) #1
mfactor2 = 1
mj_add = (mfactor1*np.exp(1j*(h-a)*temp)-mfactor2*1)/ (k+N) #N -1
# Vector m_s
ms = np.concatenate([[mj_0], mj_minus, [0], np.flip(mj)]) #2N
# Vector m_c
mc = np.concatenate([np.flip(mj_add), [mj_N], np.flip(mj), [mj_0]]) #2N

# FFT algoritm
fftms = fft(ms) #2N
fftmc = fft(mc) #2N
sgnvector = np.ones(2*N)
sgnvector[1::2] = -1 #2N


if M > 1:
    for m in range(M - 1, 0, -1):

        uj = cf*Uk

        # Vector u_s
        us = np.concatenate([uj, np.zeros(N)])

        # Matrix-vector multiplication M_s *u with
        # the help of the FFT algorithm
        fftu = fft(us)
        Msu = ifft(fftms * fftu)[0:N]
        # Matrix-vector multiplication M_c*u
        # with the help of the FFT algorothm
        Mcu = ifft(fftmc* sgnvector* fftu)[0:N]
        Mcu = np.flip(Mcu)
        # Fourier cosine coefficients option value Uk(t_m)
        Uk = (np.exp(-r*dt)/np.pi)*(np.imag(Msu+Mcu))
```

```
            Uk[0] = 0.5*Uk[0]


        Recf = (np.exp(-0.5*sig**2*dt*omega**2)
                *np.cos(omega*((r -0.5*sig**2)*dt+x-a)))
        # Option value
        U = np.exp(-r*dt)*2/(b-a)*K*np.dot(Uk, Recf)

        U_Rich.append(U)
        MRc +=1

    # 4-point Richardson extrapolation scheme (with k0=1/2, k1=1, k2=3/2)
    U_Rich = np.array(U_Rich)
    U = 1 / (5 - 3 * np.sqrt(2)) * (8 * U_Rich[3:] - (6 * np.sqrt(2) + 4)
        * U_Rich[2:-1] + (3 * np.sqrt(2) + 2) * U_Rich[1:-2] - U_Rich[:-3])

    return U
```

9 Appendix

Definition 9.1. *Let the price of an option be $c(t,x)$, the asset price x and $t \in [0,T)$. The solution to the* ***Black-Scholes-Merton*** *partial differential equation for some continuous function $c(t,x)$ is*

$$c_t(t,x) + rxc_x(t,x) + \frac{1}{2}\sigma^2 x^2 c_{xx}(t,x) = rc(t,x), \tag{74}$$

for all $x > 0$ (Shreve. P. 147).

Definition 9.2. ***The Fast Fourier transform*** *is an algorithm used to compute the discrete Fourier transform (Heckbert, 1998).*

We know that the Fourier transform is the characteristics function in probability.

Definition 9.3. ***The discrete Fourier transform*** *for a sequence $\{a_n\}_{n=0}^{N-1}$ is:*

$$A_k = \sum_{n=0}^{N-1} \exp\left\{-i\frac{2\pi}{N}kn\right\} a_n, \tag{75}$$

for $k = 0, ..., N-1$ (Heckbert).

Definition 9.4. ***The inverse discrete Fourier transform*** *for a sequence $\{a_n\}_{n=0}^{N-1}$ is:*

$$A_k^{-1} = \frac{1}{N}\sum_{n=0}^{N-1} \exp\left\{-i\frac{2\pi}{N}kn\right\} a_n, \tag{76}$$

for $k = 0, ..., N-1$ (Delgutte, Greenberg, 1999. P. 10).

Definition 9.5. *The idea of the* ***Richardson extrapolation*** *is to find a better approach using the following approaches. In general, we could use* u_h *and* $u_{\frac{h}{2}}$ *to find an approach of* $\hat{u}$ *by:*

$$\hat{u} = \alpha \cdot u_h + \beta u_{\frac{h}{2}}, \tag{77}$$

for $h \neq 0$*. For example the four-point Richardson extrapolation scheme for the first order derivative*

$$f'(x_0) \approx u_h = \frac{f(x_0 + h) - f(x_0 - h)}{2h}. \tag{78}$$

Then we can find $\hat{u}$ *by combining* u_{2h} *and* u_h*, such that:*

$$\hat{u} = -\frac{1}{3}u_{2h} + \frac{4}{3}u_h = \frac{-\frac{1}{4}f(x_0 + h) + 2f(x_0 + h) - 2f(x_0 - h) + \frac{1}{4}f(x_0 - 2h)}{3h}, \tag{79}$$

(van Leeuwen).

Definition 9.6. *Let* $a = (a_1, a_2, ..., a_n)$ *and* $x = (x_1, x_2, ..., x_n)$ *be two vectors of length* n*. Their* ***circular convolution*** $y = a \cdot x$ *is a vector of length* n *such that:*

$$y = a \cdot y \quad \Longleftrightarrow \quad y = \sum_{l=0}^{n-1} a_{k-l} x_l, \tag{80}$$

where the indices in the sum are evaluated modulo n *and* $k < l$ *(Bamieh, 2022).*