

FINANCIAL MATHEMATICS TEAM CHALLENGE

A collection of the four reports from the 2023
Financial Mathematics Team Challenge.



AIFMRM

AFRICAN INSTITUTE OF FINANCIAL MARKETS AND RISK MANAGEMENT



Preamble

One of the key aims of the FMTC is for South African postgraduate students in Financial and Insurance Mathematics to have the opportunity to focus on a topical, industry-relevant research project, while simultaneously developing links with international students and academics in the field. An allied objective is to bring a variety of international researchers to South Africa to give them a glimpse of the dynamic environment that is developing at UCT in the African Institute of Financial Markets and Risk Management. The primary goal, however, is for students to learn to work in diverse teams and to be exposed to a healthy dose of fair competition.

The Eighth Financial Mathematics Team Challenge was held from the 10th to the 20th of July 2023. The challenge brought together four teams of Masters and PhD students from Austria, Switzerland, Lesotho, Zambia, South Africa and the UK to pursue intensive research in Financial Mathematics. Each team worked on a distinct research problem over the twelve days. Professional and academic experts from Switzerland, Iceland, Australia, South Africa, and the UK individually mentored the teams; fostering teamwork and providing guidance. As they have in the past, the students applied themselves with remarkable commitment and energy.

This year's research included topical projects on (a) Bayesian Model Averaging Applied to Implied Expected Signature Models, (b) Physics-informed Neural Networks for Option Pricing and Hedging, (c) Sequential Monte Carlo for Index Tracking with Transaction Costs, and (d) South Africa's Carbon Opportunity. These were either proposed directly by our academic/industry partners or chosen from areas of current relevance to the finance and insurance industry. In order to prepare the teams, guidance and preliminary reading was given to them a month before the meeting in Cape Town. During the final two days of the challenge, the teams presented their conclusions and solutions in extended seminar talks. The team whose research findings were adjudged to be the best was awarded a floating trophy. Each team wrote a report containing a critical analysis of their research problem and the results that they obtained. This volume contains these four reports, and will be available to future FMTC participants. It may also be of use and inspiration to Masters and PhD students in Financial and Insurance Mathematics.

FMTC IX, which will take place in July 2024, is already being organised!

David Taylor, University of Cape Town

Andrea Macrina, University College London & University of Cape Town

Contents

1. **N. Schwellnus, K. Schwulst, A. Timbana, J. van der Merwe**
Bayesian Model Averaging Applied to Implied Expected Signature Models
2. **K. Davies, A. Motloutsi, N. Mthwazi, H. Wutte**
Physics-informed Neural Networks for Option Pricing and Hedging¹
3. **L. Hamilton-Russell, T. O'Callaghan, F. Primavera, D. Savin**
Sequential Monte Carlo for Index Tracking with Transaction Costs
4. **E. Hampwaye, S. Hlalukana, F. Krach, N. Rateele**
South Africa's Carbon Opportunity

¹Winning team of the eighth Financial Mathematics Team Challenge

Bayesian Model Averaging Applied to Implied Expected Signature Models

TEAM 1

NEIL SCHWELLNUS, University of Cape Town
KARA SCHWULST, University of Cape Town
ALFONSO TIMBANA, University of Cape Town
JUSTIN VAN DER MERWE, University of Cape Town

Supervisors:

CHRISTA CUCHIERO , University of Vienna
JOSEF TEICHMANN, ETH Zürich

Contents

1	Introduction	4
2	Literature review and background	7
2.1	Background on Signature Literature	7
2.2	Background to Financial Models	7
2.2.1	Affine and Polynomial Models	7
2.2.2	Heston Model	8
2.2.3	Bachelier Model	8
2.3	Bayesian Averaging	8
3	Foundations of signatures	10
3.1	The signature of a path	10
3.2	The Shuffle Product	11
3.3	The Universal approximation property	11
3.4	Pricing of Signature Payoffs	12
4	Market Implied Expected Signature	14
4.1	SABR Model	14
4.2	Methodology	15
4.3	Results	15
5	Expected Signature Under Polynomial Diffusion Models	17
5.1	Methodology	18
5.2	Results	19
6	Bayesian Model Averaging	21
6.1	Bachelier-Heston Type Model	21
6.2	Model Robustness Diagnostics	22
6.3	Methodology	24
6.4	Simulated Market Data Results	25
6.4.1	Variance Swap	25
6.4.2	Call Prices	26
6.5	Real-World Market Data Results	27
6.5.1	Data	27
6.5.2	Bayesian Averaging Applied to Real-World Market Data	28
7	Recommendations and Limitations	31

8	Conclusions	32
9	Appendix	36
9.1	Results of Expected Signatures for Uncorrelated Brownian Motion	36
9.2	Bayesian Averaging in Simplified Terms	37
9.3	Black-Scholes Formula	38
9.4	Feller Condition	38
9.5	Breeden-Litzenberger formula for variances swaps in Bachelier- type models	38

1 Introduction

When faced with limited market price data for a specific category of financial products, it becomes essential to find techniques to effectively exploit this information to determine prices of other financial products within an arbitrage-free market. Inspired by the approach of Arribas et al. (2020) and Cuchiero et al. (2023a), we use signature payoffs with the intention of applying them for a non-standard calibration technique based on so-called Bayesian model averaging (BMA).

The problem of model calibration is frequently encountered in finance, where selecting appropriate models based on data from the derivative market is crucial. These models serve various purposes, such as risk management and pricing of other more exotic derivatives. To solve the calibration problem, we shall pursue an approach which differs substantially from the standard least-square minimization between market prices and model prices. We derive a specific market-based quantity known as the market-implied expected signature. These market-implied prices of signature payoffs, independent of any specific model, enter into the likelihood function in our Bayesian model averaging. Thus, this allows us to select only the *cloud of models* whose expected signature is close to the market-implied expected signature.

Signature payoffs are a collection of path-dependent derivatives characterized by iterated integrals. Due to their nature, signature payoffs encapsulate a wealth of information encompassing approximations of all possible path dependent payoffs. In mathematical terms, let $S := (S_t)_{t \geq 0}$, $t \in [0, T]$ be a stochastic process, representing the price of the underlying, where t represents time and which can be modelled in either discrete or continuous time. The derivative market exhibits different standard payoffs. For example, the payoff of a vanilla call option, at maturity T_i and strike K_j , is given by:

$$F = (S_{T_i} - K_j)^+.$$

The market data provides a price, p , of such a derivative. This market price should then be reproduced by models such that:

$$\mathbb{E}_{\mathbb{Q}}[F] \approx p,$$

where $\mathbb{Q}$ denotes a pricing measure. In our case, we ignore the effects of interest rates, as well as market frictions. Examples of such frictions include bid-ask spreads, transaction costs and price impacts. Via market-implied expected signature and Bayesian averaging, we want to choose a cloud of models S which matches the observed market prices, p , for the payoffs, F , well.

The standard model selection approach builds on a parametric class of models and chooses one specific parameter set in an effort to reproduce the set of prices ob-

served on the market. However, there are flaws evident in such an approach. Firstly, the selected model class may be incorrect, which causes difficulty when predicting the assets' future behaviour (see Duembgen and Rogers (2014)). Secondly, numerical difficulties may arise due to Monte Carlo pricing in each of the optimization steps. Therefore, in this report, we attempt a Bayesian model averaging approach.

Bayesian model averaging (BMA) is an expanded approach to Bayesian inference that goes beyond traditional methods. In BMA, parameter uncertainty is not only captured through the prior distribution, but it also incorporates model uncertainty. By applying Bayes' theorem, BMA enables the derivation of posterior distributions for both model parameters and the models themselves. This framework facilitates direct model selection, as well as combined estimation and prediction, providing a comprehensive approach to addressing uncertainty in both parameter estimation and model choice Fragoso et al., 2018. Hence, BMA allows for a direct combination of models to obtain combined parameter estimates or predictions.

In a nutshell our approach can be summarized as follows:

- We derive market-based quantities referred to as market-implied expected signatures from prices of standard derivatives in a purely regression based way.
- We use these market-implied expected signatures, because signature is an easy to handle, model independent basis for all path dependent derivatives.
- We exploit that signature payoffs can be explicitly priced in the large polynomial class containing most of the models used in finance.
- We use market implied expected signature as indicators for the quality of a cloud of parametric models of polynomial type.
- We construct this cloud of likely models in a Bayesian sense and average over it to price and potentially hedge. This is purely integration based, optimisation-free and takes model uncertainty into account. Moreover, recalibration by updating the market information can be done consistently without switching each day to a new model.

The structure of this report is as follows. Section 2 provides an overview of previous research conducted on signatures and polynomial diffusion models, as well as some background to the different models and methods employed in this report. Section 3.1 outlines the mathematical background on signatures and pricing of signature payoffs. In Section 4, the methodology of the first task is presented, which is how to obtain market-implied expected signatures from available derivatives'

data. This is essentially a linear regression task, namely regressing signature payoffs on standard (call) payoffs. Section 5 provides the details and results of task two, which involves the calculation of the expected signature under a polynomial diffusion model. The third and final task, namely Bayesian model averaging applied to implied expected signature models, is described in Section 6. This will be accompanied by the final results obtained from this task. Recommendations and limitations will be provided in Section 7. Finally, concluding remarks will be stated in Section 8.

2 Literature review and background

2.1 Background on Signature Literature

Original studies of iterated integrals were conducted by Chen (1957), where smooth piece-wise paths were considered. Extensions of this to bounded variation and rougher paths have been explored in Lyons (1998), where the theory of rough paths has been introduced. This is a branch of analysis where signatures play an important role.

There are also many practical applications of signatures. The exploration into applications in the area of machine learning and time series analysis was done by Chevyrev and Kormilitzin (2016) and Levin et al. (2013), and studies in the area of finance by Gyurkó et al. (2013). More recently there have been some papers on the methods of kernalization of the signature, see e.g. Király and Oberhauser (2019). Furthermore Boedihardjo et al. (2020) study tail asymptotics of signatures for rough paths. Signatures have also been utilised in the approximation of solutions to Stochastic Differential Equations (SDEs) by Kusuoka (2004) and Gyurkó and Lyons (2010). So-called signature-SDEs are explored in Cuchiero et al. (2023b) where the perspective is taken from affine and polynomial processes. This was inspired by Arribas et al. (2020) who developed an alternative to so-called neural SDEs by introducing a specific sig-SDE model for quantitative finance. This was further developed in Cuchiero et al. (2022) and Cuchiero et al. (2023a). These models offer a new approach to calibration and pricing of exotic non-linear financial products.

2.2 Background to Financial Models

2.2.1 Affine and Polynomial Models

Affine and polynomial processes give rise to an abundance of stochastic models, both in finite and infinite dimensions. Examples include Ornstein-Uhlenbeck processes, the Black-Scholes model (Black and Scholes, 1973), the Heston model (Heston, 1993), Lévy processes and curve models. Affine and polynomial processes can be described as a specific class of time-homogeneous Markov processes (Cuchiero et al., 2012) which exhibit highly useful tractability properties that we shall exploit subsequently. Applications of these processes in the area of mathematical finance include

- Pricing of options,
- Term structure of interest rates,
- Credit risk,

see e.g. Duffie et al. (2003) for further details.

2.2.2 Heston Model

The Heston model was introduced by Steven Heston in 1993 (Heston, 1993). This model is a stochastic volatility model and is more flexible for pricing of derivatives than the standard Black-Scholes model (Heston, 1993). Under the physical measure, the asset price in this model follows the diffusion

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t dW_t, \quad (1)$$

with some parameter μ and W a Brownian motion. The dynamics of the variance process ν are given by a Cox-Ingersoll-Ross (Cox et al., 1985) process of the form

$$d\nu_t = \kappa(\theta - \nu_t)dt + \alpha\sqrt{\nu_t}dB_t,$$

where B is another Brownian motion correlated with W and κ, θ, α are nonnegative parameters.

2.2.3 Bachelier Model

Louis Bachelier, regarded as the father of option pricing theory, developed the Bachelier model (Bachelier, 1900). Bachelier realised that the underlying distribution of stock prices needed to be understood in order to price option contracts (Sullivan and Weithers, 1991). A normal distribution was derived for these movements, by making use of the central limit theorem of the successive price changes. Modulo a constant volatility this limit then corresponds to a Brownian motion. The Black-Scholes and Bachelier model share conceptual similarities for option pricing. One difference is that the Bachelier model can produce negative prices, but their probability can be made small by choosing appropriate starting values. The Bachelier model dynamics for the price of an asset are described as

$$dS_t = \sigma_N dW_t,$$

where σ_N represents the volatility and W a standard Brownian motion.

2.3 Bayesian Averaging

Bayesian model averaging (BMA) provides a robust framework for incorporating prior information and updating it with data, leading to improved forecasts and more accurate inference. For this reason, BMA has seen growing applications including energy (see, Zhang and Yang (2015)) and commodity forecasting and pricing (see, Amrouk and Heckeilei (2020)) and as well as mixture models for asset prices in spirit of (Duembgen and Rogers, 2014).

Hinne et al. (2020) and Steel (2020) summarize some of the advantages of BMA approaches. Firstly, BMA addresses overconfidence by considering model uncertainty, which is often underestimated when a single model is selected. Furthermore, BMA acknowledges the uncertainty associated with model choice ensuring a more robust analysis. Secondly, BMA provides optimal predictions under various loss functions, even though the true model may be unknown. This approach mitigates the errors caused by inconsistent model identification. Thirdly, BMA avoids the rigid all-or-nothing approach of classical hypothesis testing by retaining model uncertainty throughout the inference process, allowing for a more nuanced analysis. Additionally, BMA gracefully updates estimates and model weights as new data accumulate, preventing sudden shifts in estimates and reducing variance across experiments. However, this comes at the expense of assigning nonzero probabilities to potentially incorrect models. Lastly, BMA avoids the incoherence of resurrecting previously rejected models based on new data, ensuring a more coherent and adaptive approach to model selection.

3 Foundations of signatures

3.1 The signature of a path

A path is a mapping from an interval $[a, b]$ to $\mathbb{R}^d$. The signature of a path captures important geometric and analytical properties of the path (Chevyrev and Kormilitzin, 2016), by estimating numerous path-dependent quantities (Gyurkó et al., 2013). This section will delve into the basic notations, definitions and properties of signatures. These are adapted from Cuchiero et al. (2022). The n -th tensor product of $\mathbb{R}^d$ is represented by

$$(\mathbb{R}^d)^{\otimes n} := \mathbb{R}^d \otimes \dots \otimes \mathbb{R}^d \text{ for } n \in \mathbb{N}_0.$$

We introduce the extended tensor algebra as

$$T((\mathbb{R}^d)) := \{\mathbf{a} := (a_0, \dots, a_n, \dots) : a_n \in (\mathbb{R}^d)^{\otimes n}\}.$$

Likewise, the truncated tensor algebra of order N is

$$T^N(\mathbb{R}^d) := \{\mathbf{a} \in T((\mathbb{R}^d)) : a_n = 0, \forall n > N\},$$

with the tensor algebra $T(\mathbb{R}^d) := \bigcup_{N \in \mathbb{N}} T^N(\mathbb{R}^d)$.

Let X be a path of bounded variation denoted $X : [a, b] \rightarrow \mathbb{R}^d$. The increment of the i -th coordinate X at time $t \in [a, b]$ is denoted as

$$S(X)_{a,t}^i = \int_{a < s < t} dX_s^i = X_t^i - X_a^i.$$

The double-iterated integral for any pair $i, j \in \{1, \dots, d\}$ is defined as

$$S(X)_{a,t}^{i,j} = \int_{a < s < t} S(X)_{a,s}^i dX_s^j = \int_{a < r < s < t} dX_r^i dX_s^j.$$

In the aforementioned formulas, $S(X)_{a,s}^i$ and X_s^j are real-valued paths (Chevyrev and Kormilitzin, 2016). This can be extended by iterating recursively through a collection of indices $i_1, \dots, i_k$ for integers $k \geq 1$. The iterated integrals can be written as

$$S(X)_{a,t}^{i_1, \dots, i_k} = \int_{a < t_k < t} \dots \int_{a < t_1 < t_2} dX_{t_1}^{i_1} \dots dX_{t_k}^{i_k}.$$

Definition 3.1. Let $\mathbb{X}_{a,b}$ indicate the signature of $X : [a, b] \rightarrow \mathbb{R}^d$. Thus, $\mathbb{X}_{a,b}$ represents the infinite sequence of iterated integrals of X

$$\mathbb{X}_{a,b} = \left(1, S(X)_{a,b}^1, \dots, S(X)_{a,b}^d, S(X)_{a,b}^{1,1}, S(X)_{a,b}^{1,2}, \dots\right).$$

If we consider a path X on the interval $[0, t]$, we usually only write $\mathbb{X}_t$ for its signature.

The set of words, $\mathcal{I}$, is the set of multi-indices, through which the superscripts moves

$$\mathcal{I} = \{(i_1, \dots, i_k) \mid k \geq 1, i_1, \dots, i_k \in \{1, \dots, d\}\}.$$

The signature consists of lower and higher-order terms, which can be interpreted in the following manner.

- **Lower Order Terms:** These terms are at most of order three and exhibit relatively simple geometric interpretations in terms of, increments, areas and volumes. The first-order terms are the increments of the elements

$$X_{a,b}^{(i)} = X_b^i - X_a^i \text{ for } i = 1, \dots, d.$$

- **Higher Order Terms:** The terms represent the generalised polynomials of the paths.

3.2 The Shuffle Product

An important property of the signature is the ability to represent the product of $S(X)_{a,b}^{i_1, \dots, i_n}$ and $S(X)_{a,b}^{j_1, \dots, j_m}$ as a sum of a series of $S(X)_{a,b}$ (Chevyrev and Kormilitzin, 2016). This series is only dependent on the set of multi-indices, I and J . We recall the following definition from Cuchiero et al. (2023b).

Definition 3.2. For a multi-index $I = \{i_1, \dots, i_n\}$, let the basis elements of $(\mathbb{R}^d)^{\otimes n}$ be represented as $e_I := e_{i_1} \otimes \dots \otimes e_{i_n}$ where e_i denotes the canonical basis of $\mathbb{R}^d$. Let now $J = \{j_1, \dots, j_m\}$ be another multi-index. Then the shuffle product is defined as

$$e_I \sqcup e_J = (e_{I'} \sqcup e_J) \otimes e_{i_n} + (e_I \sqcup e_{J'}) \otimes e_{j_m}.$$

Thus, we are able to obtain a linear combination of higher-order terms from the product of two terms (Chevyrev and Kormilitzin, 2016).

3.3 The Universal approximation property

In view of the so-called *universal approximation property*, linear functionals of the signature play an important role. We denote them by

$$L(\mathbb{X}_t) = \sum_{0 \leq |I| \leq n} l_I \langle e_I, \mathbb{X}_t \rangle \text{ for } n \in \mathbb{N}, l_I \in \mathbb{R}.$$

The proof of this universal approximation result relies on the Stone-Weierstrass Theorem. The following properties are required to apply it:

1. **Point-separation:** the set of all linear functionals of signatures of a time extended path determine the path uniquely.

2. Algebra property: the product of two linear functionals of the signature is again a linear functional of the signature.

Due to these properties the Stone-Weierstrass theorem implies a uniform approximation of continuous (with respect to certain variation norms) paths functionals via linear functionals of the signature on compact sets of paths.

Note that the second point is a result of the following shuffle product theorem derived by Ree (1958).

Theorem 3.3. Fix two multi-indices $I = (i_1, \dots, i_n)$ and $J = (j_1, \dots, j_m)$. Then

$$\langle e_I, \mathbb{X}_t \rangle \langle e_J, \mathbb{X}_t \rangle = \langle e_{I \sqcup J}, \mathbb{X}_t \rangle.$$

3.4 Pricing of Signature Payoffs

In this section, the notion of signature payoffs is introduced. A signature payoff is simply defined as a linear functional of signature, i.e. linear combination of iterated integrals. This encompasses a large class of path dependent derivatives as it utilizes iterated integrals against a certain path (Lyons et al., 2019). The definition below introduces a signature payoffs as presented in Cuchiero et al. (2022)

Definition 3.4. Suppose that the price process, S , is given by a continuous semi-martingale on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. A payoff $F : \Omega \rightarrow \mathbb{R}$ is said to be a signature payoff with maturity t if there exists $m \in \mathbb{N}$, and $\ell := \{\ell_\emptyset, \ell_J : 0 < |J| \leq m\}$, such that

$$F := \ell_\emptyset + \sum_{0 < |J| \leq m} \ell_J \langle e_J, \widehat{S}_t \rangle,$$

where $\widehat{S}$ denotes the signature of $\widehat{S}_t := (t, S_t)$. It is thus nothing else then a linear functional of $\widehat{S}_t$.

Hence, the computation of the expected signature of $\widehat{S}_t$ under a risk neutral measure is important for pricing purposes of the signature-payoffs.

Using the above universal approximation property, signature payoffs allow for the approximation of general (continuous) path-dependent payoffs, as long as the paths of $\widehat{S}$ determining the payoffs belong to a compact set with high probability. The following theorem makes this precise (see (Lyons et al., 2019)), assuming that Ω is the canonical path space and further conditions on the probabilistic model for $\widehat{S}$ (which we do not state here).

Theorem 3.5. : Let $H : \Omega \rightarrow \mathbb{R}$ be a continuous (with respect to a certain variation norm) payoff. Let $\epsilon > 0$. Then, there exists a compact set of paths $K_\epsilon \subset \Omega$ and a signature payoff F such that

1. $\mathbb{Q}(\hat{S}_{[0,t]} \in K_\epsilon) > 1 - \epsilon$ for all risk-neutral measures $\mathbb{Q}$;
2. $|H - F| < \epsilon$ for all $\hat{S}_{[0,t]} \in K_\epsilon$.

4 Market Implied Expected Signature

In Gyurkó et al. (2013) signature-based methods were developed in an attempt to improve the standard approach for model selection. The idea behind this approach is to calculate the signature of the underlying models, as this is able to classify the underlying features of the data by utilising only a small number of coefficients. Later on in (Lyons et al., 2019) signature payoffs, i.e. linear functions of the signature of the time augmented underlying as defined above, are used to approximate the payoffs of possibly exotic options.

We go here the other way round and *approximate signature payoffs by linear functions of standardised payoffs*, such as calls or puts. Therefore, the initial step is to find a good way to move from the prices of the standardised derivatives to implied prices of signature payoffs.

Our solution is on the level of trajectories, i.e. we try to find linear combinations of the given standardised payoffs $F_i(\omega)$ (in our case standard calls), in order to approximate the signature payoffs as follows:

$$\sum_{i=1}^n l_i^{\mathcal{I}} F_i(\omega) \approx \mathbb{X}_{\mathcal{I}}^{\mathcal{I}}(\omega). \quad (2)$$

Here, ω are model generated trajectories of the financial market and $\mathbb{X}^{\mathcal{I}}(\omega)$ denotes the signature component corresponding to the multi-index $\mathcal{I}$ of the involved model quantities $X(\omega)$, precisely specified below. This reduces to a regression problem, where $l_i^{\mathcal{I}}$ are the unknown coefficients which need to be determined for each of the considered multi-indices $\mathcal{I}$. Consequentially, the first task involves performing a linear regression.

4.1 SABR Model

In order to tackle the above problem, we need to choose a set of training trajectories, where the linear equations should hold as well as possible. To generate the training data we choose the Stochastic-Alpha-Beta-Rho (SABR) model, which is a two-factor stochastic volatility model. This model was introduced by Hagan et al., 2002. The process for the price S , and the volatility σ_t , is given by

$$dS_t = S_t^{\beta} \sigma_t dW_t,$$

$$d\sigma_t = \alpha \sigma_t dB_t.$$

The parameter β relates to elasticity (see (Choi and Wu, 2021)). The instantaneous correlation between the two Brownian motions, W and B is

$$d\langle B, W \rangle_t = \rho dt.$$

This model is a popular stochastic volatility model as it is able to fit to implied volatility as observed on the market (see (Hagan et al., 2002)) relatively well despite being quite parsimonious in terms of parameters. This is crucial for risk management.

4.2 Methodology

To compute the regression coefficients in (2), calculating the observations $(X_{t_i})_{i=1}^N$ is the first step. For this, the SABR model is used where we choose a range of appropriate parameters in line with typical market trajectories. We use the simulated paths to compute the signature $\mathbb{X}$ of X , where

$$X_t = (S_t, V_t, t).$$

Here, V_t is the instantaneous variance, in the SABR model given by $V_t = \sigma_t^2 S_t^{2\beta}$. Based on the observations $(X_{t_i})_{i=1}^N$, the signature components, $(\mathbb{X}_{t_i})_{i=1}^N$, are computed. In efforts to compute this, the available packages tested were signatory by Kidger and Lyons (2020) and Reizenstein and Graham (2018) iisignature in Python. The latter is then used throughout.

4.3 Results

In the simulation, the initial value of S_0 is set to 1.0. The ranges of the input parameters are

$$\begin{aligned}\beta &\in [0.1, 1], \\ \sigma &\in [0.15, 0.25], \\ \alpha &\in [0.2, 0.4], \\ \rho &\in [-0.8, 0.2].\end{aligned}$$

The specific set of parameters that is finally chosen is

$$\{S_0, \sigma_0, \alpha, \beta, \rho\} = \{1, 0.15, 0.25, 0.1, -0.5\}.$$

Using the iisignature package, the signature components are computed up to order 4, for each path, with 100000 paths being generated. This package omits the first value of the signature, which is the signature corresponding to the empty set and is always 1. Consequently, we appended the first value to the signature for every path. For the purpose of performing linear regression, both the linear and ridge regression packages in Python were tested. A value of 0.05 was used for the regularization parameter in the ridge regression. The linear coefficient results for linear regression were very large. As a result, we decide to use Ridge regression, in order to implement a penalty term, and regularize our regression procedure.

To check the accuracy of the linear regression, different call payoffs of the form

$$F_i = (S_{T_i} - K_i)^+$$

are computed where S_T is calculated using the SABR model simulations. We utilise 10 strikes, K , in the range $[0.6, 1.5]$ and 12 maturities. These are then multiplied against the computed regression coefficients and compared to the computed signature components. Figure 1 provides a plot of the expectations of these quantities to assess predictability.

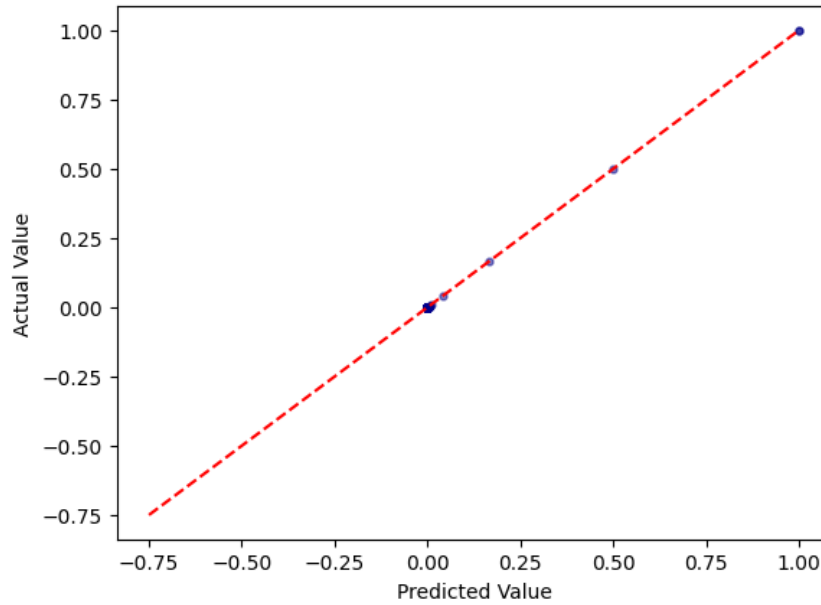


Figure 1: Sanity check for the computation of expected signatures

5 Expected Signature Under Polynomial Diffusion Models

This section delves into the model class of polynomial diffusion models. This model class will be used to compute the expected signatures. The choice of this model class is due to their flexibility and analytical tractability. Indeed, we can compute expected signatures in this model class in a highly tractable way. Define $(Y_t)_{t \geq 0}$ to be a polynomial diffusion process with the dynamics of the process given by

$$dY_t = b(Y_t)dt + \sqrt{a(Y_t)}dW_t, \quad Y_0 = y_0.$$

Here, $b : \mathbb{R}^d \rightarrow \mathbb{R}^d$, $a : \mathbb{R}^d \rightarrow \mathbb{R}$, where a is a positive semi-definite matrix, and $(W_t)_{t \geq 0}$ a d -dimensional standard Brownian motion. We require b to be an affine function and a to be a quadratic function. We denote the signature by $\mathbb{Y}$. Subsequently, we derive the expected signature of $(Y_t)_{t \geq 0}$, where the derivation is adapted from Cuchiero et al. (2022). Define $b_i(y)$ and $a_{ij}(y)$ as

$$b_i(y) = b_i^0 + \sum_{k=1}^d b_i^k y_k,$$

$$a_{ij}(y) = a_{ij}^0 + \sum_{k=1}^d a_{ij}^k y_k + \sum_{h,k=1}^d a_{ij}^{hk} y_k y_h,$$

where $a_{ij}^{hk} = a_{ij}^{kh}$.

We can write these in terms of signatures

$$b_i(y_t) = \langle \mathbf{b}_i, \mathbb{Y}_t \rangle,$$

with $\mathbf{b}_i$ defined as

$$\mathbf{b}_i = \left(b_i^0 + \sum_{k=1}^d b_i^k y_0^k \right) e_\emptyset + \sum_{k=1}^d b_i^k e_k.$$

To represent a_{ij} in terms of signatures, we write

$$a_{ij}(y_t) = \langle \mathbf{a}_{ij}, \mathbb{Y}_t \rangle$$

$$\mathbf{a}_{ij} = \left(a_{ij}^0 + \sum_{k=1}^d a_{ij}^k y_0^k + \sum_{k,h=1}^d a_{ij}^{kh} y_0^k y_0^h \right) e_\emptyset + \sum_{k=1}^d \left(a_{ij}^k + 2 \sum_{h=1}^d y_0^h \right) e_k + \sum_{k,h=1}^d a_{ij}^{kh} e_k \sqcup e_h,$$

For $I = \{i_1, \dots, i_n\}$ with $|I| \leq n$. The truncated signatures can be written as

$$d\langle e_I, \mathbb{Y}_t \rangle = \langle Le_I, \mathbb{Y}_t \rangle dt + \langle e_I, \mathbb{Y}_t \rangle \left(\sqrt{a(y_T)} \right)_{i_n} dW_t,$$

with $e_{I'}$ representing the last element in I being deleted. The truncated signatures, $\mathbb{Y}^n$, are also polynomial diffusions. The operator L , termed the dual operator corresponding to $\mathbb{Y}$ (Cuchiero et al., 2023a), is defined as

$$\begin{aligned} Le_I &= e_{I'} \sqcup \mathbf{b}_{ij} + \frac{1}{2} e_{I''} \sqcup \mathbf{a}_{i_{n-1}i_n} \\ &= \sum_{|J| \leq n} \eta_{IJ} e_J. \end{aligned}$$

This is then used to compute a matrix $\mathcal{G} \in \mathbb{R}^{d_n \times d_n}$

$$\mathcal{G}_{\mathcal{L}(I)\mathcal{L}(J)} = \eta_{IJ},$$

where d_n is the dimension of the truncated signature space and $\mathcal{L} : \{I \rightarrow |I| \leq n\} \rightarrow \{1, \dots, d_n\}$ is a labelling injective function (Cuchiero et al., 2023a). One can compute the expected values of the signature components, $\mathbb{Y}_t^n$ via

$$\mathbb{E} \begin{bmatrix} 1 \\ \mathbb{Y}_T \\ \mathbb{Y}_T^2 \\ \vdots \\ \mathbb{Y}_T^n \end{bmatrix} = e^{T\mathcal{G}^*} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $\mathcal{G}^*$ the transpose of $\mathcal{G}$ and $e^{T\mathcal{G}^*}$ denotes the matrix exponential (Cuchiero et al., 2023a). This is used to compare with the market implied expected signatures and to find the parameters of b and a such that the market implied quantities are matched as closely as possible. The main benefit of this is that the expected signatures can be written as a matrix exponential of the matrix $\mathcal{G}$ which is relatively easy to compute.

5.1 Methodology

The model chosen is the SABR model, with the same dynamics as those depicted in 4.1 here with $\beta = 0$. This is used to compute $Y_t = (t, S_t, \sigma_t)$ using Monte Carlo simulation and to calculate the signature components, $\mathbb{Y}_t$, using iisignature. The expected signatures are derived by taking the expected value over the simulated the paths. In return, this is compared to the analytical formula which makes use of Sara Svaluto-Ferro's code for the computation of the expected signature of a continuous polynomial process. Before implementing the current SABR model model, a simple model of two uncorrelated Brownian motions ($\rho = 0$) is tested. This helps us become familiar with how the code works, and what inputs are required for the constant b_i^0, a_{ij}^0 , linear b_i^k, a_{ij}^k and the quadratic terms a_{ij}^{hk} for the correct execution

of the code.

The results for the implementation of the two-dimensional uncorrelated Brownian motion can be found in Section 9.1 of the Appendix.

5.2 Results

For the simulation of the data, we select the following parameter inputs into the SABR model

$$\{S_0, \sigma_0, \alpha, \beta, \rho\} = \{1, 0.15, 0.25, 0.1, -0.5\}.$$

The calibrations are performed with a maturity of one year, with $N = 100000$ Monte Carlo simulations. The calculation of the analytical formula for the expected signatures is computed using Sara Svaluto-Ferro's code (Svaluto-Ferro).

The dynamics of the model result in the following constant, linear and quadratic terms for b and a respectively are as follows

$$\begin{aligned} b_1^0 &= 1, b_i^k = 0 \text{ for } i, k = 1, 2, 3, \\ a_{ij}^0 &= 0 \text{ for } i, j = 1, 2, 3, \\ a_{ij}^k &= 0 \text{ for } i, j, k = 1, 2, 3, \\ a_{ij}^{kh} &= \begin{cases} 1 & \text{if } i = j = 1, \text{ and } k = h = 2 \\ \alpha\rho & \text{if } i = 1, 2, j = 1, 2, \text{ and } k = h = 2 \\ \alpha^2 & \text{if } i = j = 2, \text{ and } k = h = 2 \\ 0 & \text{otherwise} . \end{cases} \end{aligned}$$

The analytical value for the expected signatures are then compared against the expected signatures computed via the Monte Carlo estimates. The MSE is computed and the graphs of the analytical and Monte Carlo expected signatures are plotted.

Table 1: Mean Squared Error for the Analytical and Monte Carlo Expected Signatures

Mean Squared Error
7.1799603e-09

A graphic comparison of the expected signatures for the analytical formula against the Monte Carlo simulations can be found in Figure 3. Figure 2 displays the error values which were calculated by taking the analytically computed expected signatures minus the Monte Carlo computed expected signatures. Figure 2 depicts the error values that are randomly scattered on the residual = 0 line. The small value of the MSE, together with the graphic representations show that the analytical formula and the Monte Carlo computations are very close.

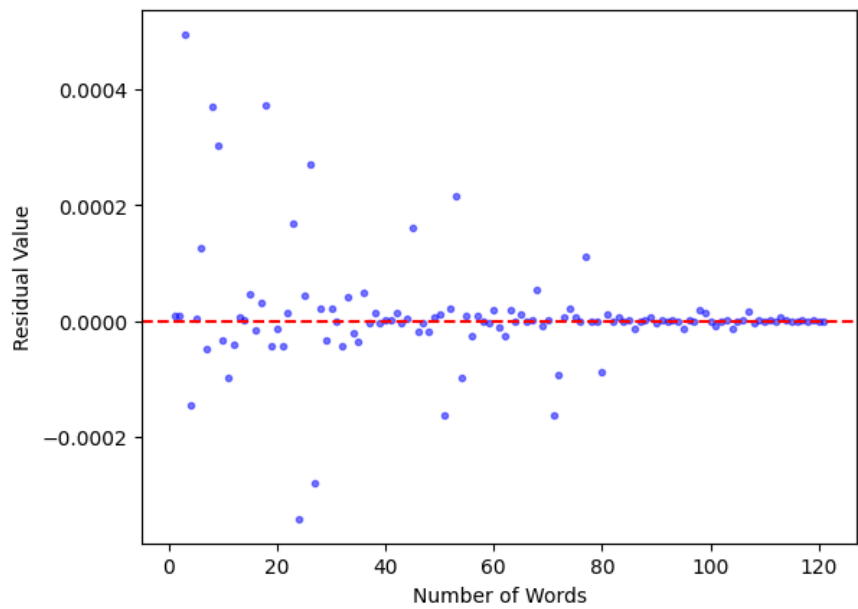


Figure 2: Sanity check for the computation of expected signatures

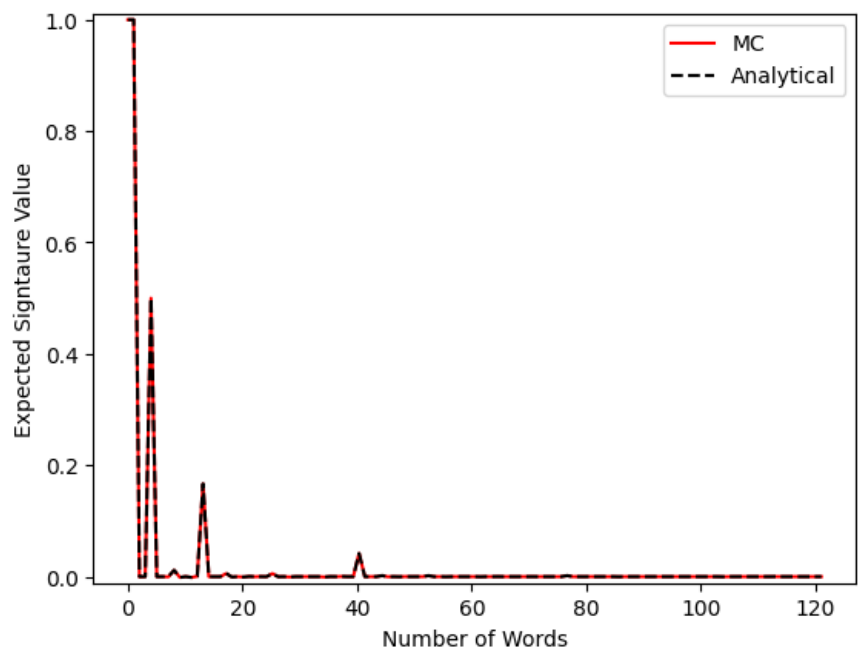


Figure 3: Sanity check for the computation of expected signatures

6 Bayesian Model Averaging

We begin with a parametric class of models (Y_t^θ) , which, in our case, is the polynomial diffusion model class. To determine the prices for the signature payoffs, we make use of the analytical formula to compute

$$\mathbb{E}^\theta[\mathbb{Y}^{\mathcal{I}}] = p_{\mathcal{I}}^\theta \text{ for } i = 1, \dots, n$$

where $\mathbb{Y}^{\mathcal{I}}$ are the signature payoffs, and $p_{\mathcal{I}}^\theta$ are the prices we obtain for them. We then use this for Bayesian Averaging

$$p = \int_{\theta} p_{\mathcal{I}_0}^\theta \frac{e^{-\frac{\sum_{\mathcal{I}} \|p_{\mathcal{I}}^\theta - p_{\mathcal{I}}\|^2}{\lambda}}}{c_\lambda} \pi(d\theta)$$

where c_λ is a normalising factor to normalise the likelihood time the prior to a probability measure, i.e. c_λ is

$$c_\lambda = \int_{\theta} e^{-\frac{\sum_{\mathcal{I}} \|p_{\mathcal{I}}^\theta - p_{\mathcal{I}}\|^2}{\lambda}} \pi(d\theta).$$

The value of the parameter λ is typically chosen to be small and the integration occurs over some prior π distribution, for example, the uniform distribution. In a slightly different setup Duembgen and Rogers (2014) assert that the key requirement is to set λ at a value such that the contribution to the log-likelihood from the moves of the underlying and from the call surface fitting errors should be of similar orders of magnitude.

The intended meaning of the quadratic term $\|p_{\mathcal{I}}^\theta - p_{\mathcal{I}}\|^2$ is that the disparity between the observed market prices, $p_{\mathcal{I}}$, and the Analytical expected signature, $p_{\mathcal{I}}^\theta$. We take an uninformed prior π , thus drawing from the multidimensional uniform distribution. The quantity $e^{-\frac{\sum_{i=1}^n \|p_{\mathcal{I}}^\theta - p_{\mathcal{I}}\|^2}{\lambda}}$ will be relatively large when the computed prices are close to the market prices and small when there is a big difference between the two. Hence, if there is a big difference between computed and observed, one just throws it away. The question is whether the outputted price, p , is realistic, in line with market prices.

6.1 Bachelier-Heston Type Model

For the purpose of simulating data, we employ the Bachelier-Heston type model following Bachelier (1900) and Heston (1993). Here, the dynamics of the underlying asset, S , and variance, V , are given by

$$dS_t = \sqrt{V_t} dW_t,$$

$$dV_t = \alpha \sqrt{V_t} dB_t + \kappa(\theta - V_t)dt,$$

$$d\langle B, W \rangle_t = \rho dt,$$

where W_t and B_t are Brownian motions with constant correlation ρ .

The dynamics of the model result in the following constant, linear and quadratic terms for b and a respectively

$$b_1^0 = 1, b_3^0 = \kappa\theta,$$

$$b_i^k = \begin{cases} 1 & \text{if } i, k = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij}^0 = 0 \text{ for } i, j = 1, 2, 3,$$

$$a_{ij}^k = \begin{cases} 1 & \text{if } i = j = 1, \text{ and } k = 2 \\ \alpha\rho & \text{if } i = 1, 2, j = 1, 2, \text{ and } k = 2 \\ \alpha^2 & \text{if } i = j = 2, \text{ and } k = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ij}^{kh} = 0 \text{ for } i, j, k = 1, 2, 3,$$

6.2 Model Robustness Diagnostics

Before implementing the Bayesian averaging approach, the robustness of our model needs to be validated. The following procedure is implemented:

1. Simulate data with a new set of parameters to calculate new payoffs.
2. The new payoffs are regressed against regression coefficients calculated with the old set of parameters, which represents the new signature components.
3. Calculate the expected signatures for the new set of signature components.
4. Compute the expected signatures using the analytical formula.
5. Compare the Monte Carlo expected signatures using the regression against the analytical expected signatures.

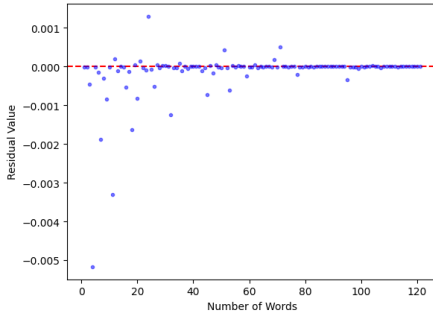
Firstly, Simulation of data is performed with initial parameters set to

$$\{S_0, V_0, \alpha, \kappa, \rho, \theta\} = \{1, 0.0225, 0.04, 1, -0.5, 0.05\}.$$

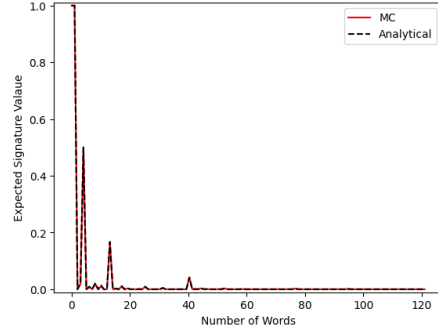
Secondly, the expected signatures are calculated using the analytical formula with a new set of parameters, Θ^{new} . This was performed separately by increasing each

parameter in Θ by 20%, decreasing each parameter in Θ by 20% and randomly increasing some and decreasing other parameters by varying percentage values, whilst still ensuring the Feller condition is satisfied. The parameters for the third test were set as

$$\{\alpha_{\text{new}}, \rho_{\text{new}}, \kappa_{\text{new}}, \theta_{\text{new}}, V_{0_{\text{new}}}\} = \{0.9\alpha, 0.85\rho, 1.2\kappa, 1.1\theta, 0.95V_0\}.$$

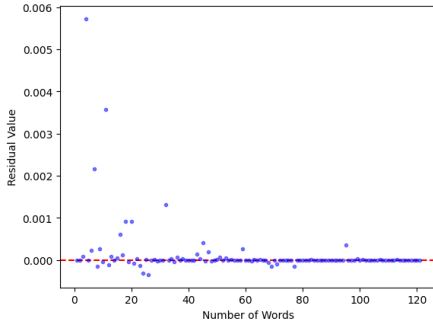


(a) Residuals over signatures for 1.2Θ

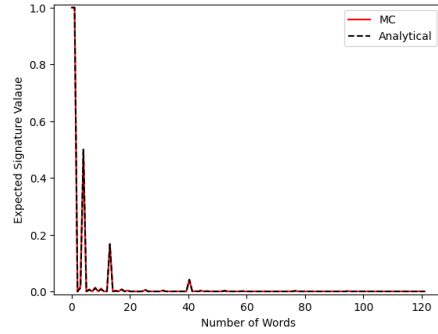


(b) MC vs Analytical Exp. Signature

Figure 4: Residuals and comparison of expected signatures for 1.2Θ



(a) Residuals over signatures for 0.8Θ



(b) MC vs Analytical Exp. Signature

Figure 5: Residuals and comparison of expected signatures for 0.8Θ

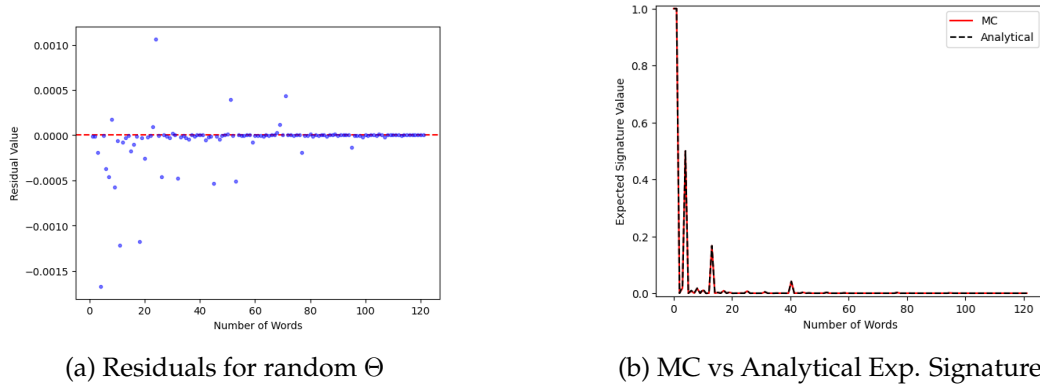


Figure 6: Residuals and comparison of expected signatures for random Θ

Based on the parameter selection Θ , it is evident from Figure 4a, 5a, and 6a that the residuals exhibit a concentration around the zero bound. Notably, when using a larger parameter set Θ , the Monte Carlo simulations tend to slightly underestimate, while with a smaller parameter set, the Monte Carlo overestimates. However, for randomly selected parameters, the Monte Carlo simulated residuals align well with expectations.

Nonetheless, the comparison between Monte Carlo and expected analytical values for the signatures, as depicted in Figure 4b, 5b, and 6b, indicates a good fit across all parameter selections. Thus, we can conclude that our model is robust, and insensitive to varying the initial parameters.

6.3 Methodology

The first step involves checking if the calibration works on simulated data, before implementing it on market data. Thus, the Bachelier-Heston type model is used to simulate the stock prices and variances, which, together with time, determine the path for the signature component computation. Our simulated market data is calculated by regressing the simulated payoffs against the linear coefficients to get the expected signature. This allows us to be able to estimate any payoff using our Bayesian Model Averaging formula and compare this to the simulated market data, $p_{\mathcal{I}}$. In order to test this, we make use of a standard call. We examine results for different ranges of initial parameters, together with varying strikes against a fixed maturity. Once this method has been fine-tuned, we are in a position to implement the market data on the model.

6.4 Simulated Market Data Results

To simulate the data with the Bachelier-Heston type model, the set of parameters

$$\{S_0, V_0, \alpha, \kappa, \rho, \theta\} = \{1, 0.0225, 0.04, 1, -0.5, 0.05\},$$

is chosen, whilst ensuring that the Feller Condition is satisfied (refer to section 9.4 in the Appendix for more information on the Feller Condition). Bayesian averaging on a variance swap and on simulated standard call prices is performed with the following set of initial parameters

Parameters	Lower End	Upper End
κ	0.5	1.5
θ	0.01	0.09
α	0.01	0.09
ρ	-0.9	-0.1
σ	0.01	0.05

Table 2: Interval of parameters for BMA model

6.4.1 Variance Swap

We initially implement variance swaps using the Bayesian averaging method. This involves utilising the variance swap, as it is one of the most liquid observable market signatures for a specific underlying asset, which we use to calculate the price using the Bayesian model averaging (BMA) approach. A variance swap in our case is defined as:

$$\mathbb{E} \left[\int_0^T (V_t - V_0) dt \right].$$

We modify the Bayesian averaging formula accordingly to represent our defined case of a variance swap. The variance swap payoff is a specific word which is calculated from the analytical expected signature. The variance swap price is denoted as $p_{I_0}^\theta$ where I_0 denotes a specific word in our expected analytical signature. The weighting parameter, λ , chosen in the BMA approach to price a variance payoff is chosen as 0.01.

To obtain the price, we apply the BMA method to a range of parameter intervals and perform Monte Carlo simulations. To balance computation time, we conduct 1000 simulations and achieve a relative error of 1% for pricing the variance swap. This indicates our ability to accurately price a variance swap using simulated data.

6.4.2 Call Prices

We have thus shown that using simulated market data, we can accurately calculate the price of a variance swap. We will proceed with calculating prices of standard call options through the BMA using simulated market data. The price of a standard call option is defined as:

$$\mathbb{E}[(S_T - K)^+].$$

In the BMA formulation, the price for a specific strike and maturity is presented by $p_{T_0}^\theta$ but no specific word is referenced from the analytical expected signature. It rather represents a price with a specific strike and maturity. The implementation of BMA to calculate standard call prices is implemented for a fixed maturity of 12 months and 10 different strikes between 0.6 and 1.5.

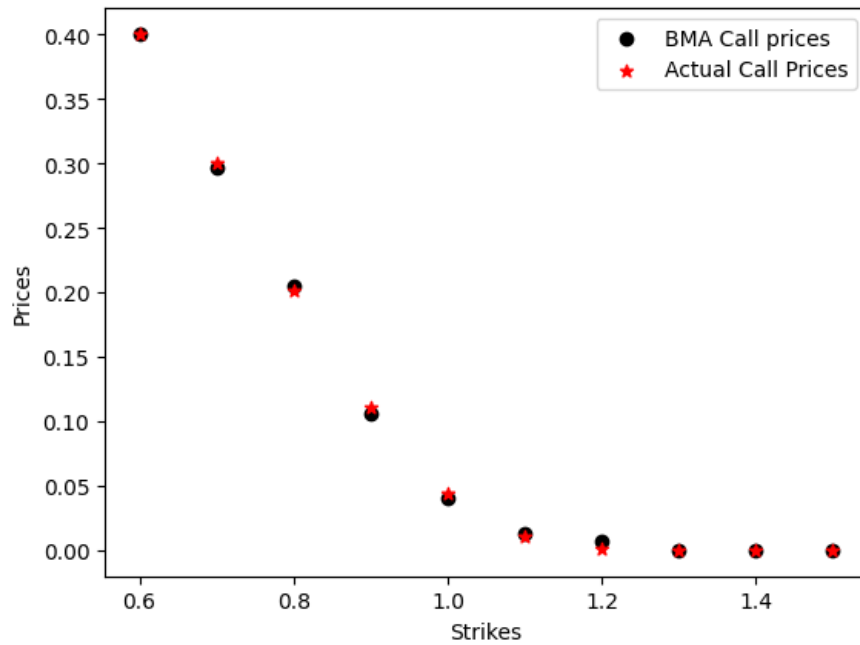


Figure 7: Comparison of BMA call prices vs Actual call prices

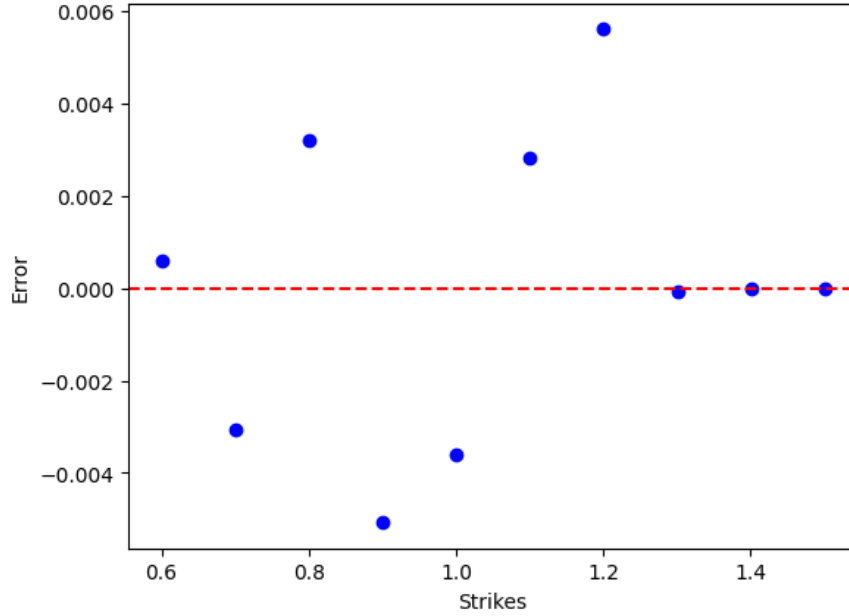


Figure 8: Comparison of BMA call prices vs actual call prices

In Figure 8, the errors for out-of-the-money options are close to zero, indicating accurate predictions. This demonstrates that the BMA call prices align closely with the actual call prices, as shown in Figure 8. We can see that the error observed for out-of-the-money options is very close to zero. The price of an out-of-the-money call option is very low and thus the absolute error between estimated and actual prices may be low even though the relative error may still be high between the respective prices. The absolute error exhibited indicate that our Bayesian Averaging model accurately estimates call prices for simulated data. Additionally, we observe the typical relationship between call prices and varying strikes.

6.5 Real-World Market Data Results

6.5.1 Data

The market data employed in the report is the S&P 500 implied volatility surface of 02/06/2021. The maturities of the data correspond to 14, 44, 58, 79, 107, 135, 170, 289, 302, 316, 352, 380, 562 and 962 days. The prices of the derivatives are computed using the Black-Scholes model (Black and Scholes, 1973), where the exact pricing formulas can be found in Appendix 9.3.

6.5.2 Bayesian Averaging Applied to Real-World Market Data

We once again remind ourselves of the Bayesian averaging approach formula:

$$p = \int_{\theta} p^{\theta} \frac{e^{-\frac{\sum_{\mathcal{I}} \|p_{\mathcal{I}}^{\theta} - p_{\mathcal{I}}\|^2}{\lambda}}}{c_{\lambda}} \pi(d\theta)$$

where c_{λ} is a normalising factor given by

$$c_{\lambda} = \int_{\theta} e^{-\frac{\sum_{\mathcal{I}} \|p_{\mathcal{I}}^{\theta} - p_{\mathcal{I}}\|^2}{\lambda}} \pi(d\theta).$$

The $p_{\mathcal{I}}$ represented in the formulation above now exhibits the real-world market data. In order to test the Bayesian averaging procedure for real-world market data we will be pricing the payoff of a standard call option. The price of a standard call option for a specified maturity and strike will now be represented by p^{θ} . Once again in the formulation $p_{\mathcal{I}}^{\theta}$ is our analytical expected signature for a particular θ parameter set.

A reasonable range of parameter intervals is chosen and λ is set to 0.01. We now apply the Bayesian averaging approach. The integral in the approach is once again calculated by estimating the integral through Monte Carlo samples for an interval of parameters. Thus, the price of our derivatives are calculated through the Bayesian averaging approach. The Bachelier-Heston type model is utilised to simulate a stock price path, and the parameter range utilized in Bayesian averaging is shown in the table below:

Parameters	Lower End	Upper End
κ	2.25	2.75
θ	0.04	0.05
α	0.175	0.225
ρ	-0.5	-0.4
σ	0.02	0.03

Table 3: Interval of parameters for BMA model

Using the interval specified above for initialised parameters, the Bayesian model averaging is implemented to price standard call options for a specific combination of strikes and maturities. We firstly analyse for an at-the-money call option for a maturity of length 107 days to analyse the effect of an increase in number of Monte Carlo samples, i.e. number of parameter sets within our specified intervals chosen. The figure below depicts the effect of increasing the number of Monte Carlo

samples and we can see that as Monte Carlo samples increase the relative error decreases and converges closer to zero. This indicates that as sample sizes increase the quality of prices obtained from the Bayesian Model Averaging approach improves.

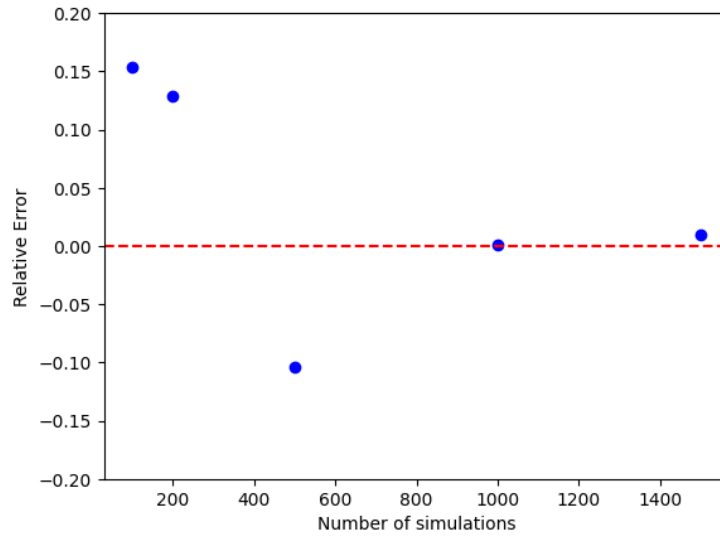


Figure 9: Relative error by number of parameter samples

We can now price standard call options for different combinations of strikes and maturities. Consequently, we implement the Bayesian averaging for a standard call option on real-world market data. The results are depicted in the table below. An initial stock price of 1 was utilized and thus strikes greater than 1 represent out-the-money and less than one represent in-the-money option prices. We can see that for in-the-money options the Bayesian Model Averaging provides accurate option prices compared to the market for all maturities. However, for out-of-the-money call options, it is yet to calculate the prices accurately when measured on a relative scale.

	BMA	Market
$T = 0.2932, K = 0.9996$	0.034319	0.034275
$T = 0.7918, K = 0.9730$	0.075321	0.073334
$T = 1.5397, K = 0.9730$	0.103595	0.103595
$T = 1.5397, K = 0.6376$	0.366302	0.362914
$T = 1.5397, K = 1.7176$	0.000112	0.025472

Table 4: Call prices for combinations of strikes and maturities

We have thus shown that the Bayesian model averaging method produces accu-

rate results for simulated data for standard call options and a variance swap. We then proceeded to implementing the method using market data and were able to produce accurate estimates for in-the-money call options for all maturities. For out-the-money call options there is still space for improvement, which could be achieved by calibrating to implied volatilities rather than prices.

7 Recommendations and Limitations

The research and methodology we have implemented through using Bayesian model averaging to implied expected signature models indicates that one can further improve this model by a number of techniques. Firstly, the computation time can be reduced with the implementation of neural networks, where a neural network can be used to calculate the analytical expected signature. This will result in an almost instantaneous calculation of the analytical expected signature which will drastically reduce computation time when implementing the Bayesian model averaging method. Consequentially, one will be able to substantially increase the number of Monte Carlo samples for the Bayesian averaging method for different parameter sets from our uninformative prior distribution. This will allow for a wider interval of initialised parameter sets to be chosen, leading to a higher accuracy of the price for a specific payoff.

Secondly, an integral aspect of the Bayesian model averaging method is the weighting parameter, λ . One would be able to tune λ in the Bayesian averaging formula in order to produce an optimal estimate for a price of a payoff as λ weights our loss function in the Bayesian averaging implementation. If λ is chosen to be too large, the accuracy of Bayesian averaging will be impacted as a large weight will be placed on prices where the difference between the analytical expected signatures and market prices is large. On the other hand, a very small λ will enlarge the exponential to infinity, resulting in an infeasible price being estimated. Therefore, tuning λ will contribute to an increased accuracy in the estimation of the price for a specific payoff.

Thirdly, one of the most liquid observable market expected signature is the variance swap. Using the Breeden-Litzenberger formula together with market data for call prices, a comparison can be made between the Bayesian approach and the calculated market variance swap prices (Breeden and Litzenberger, 1978). This can be implemented for multiple maturities which will further test the robustness of the Bayesian averaging method applied to market data, here for truly path dependent options.

Lastly, the Bayesian averaging method requires an interval of reasonable starting parameter values. Tuning the starting parameter intervals, which can be seen as a hyper-parameter optimisation, will result in a higher accuracy for calculating the prices of a specific payoff. Consequently, a high order of convergence will be obtained with a lower number of Monte Carlo samples, reducing the computation time substantially.

8 Conclusions

Let us recap the procedures we have implemented in the report. Firstly, we derived the market-implied expected signatures from standard derivative prices. This was performed via linear regression. We then calculate the expected signature under polynomial models. Using the market-implied expected signature as indicators for the quality of the polynomial models (here we implemented the Bachelier-Heston models), we obtain a cloud of models where we can implement an optimisation-free calibration, using Bayesian averaging. This allows us to reproduce accurate option prices.

In conclusion, efficient technologies for model calibration are extremely important. Our proposed method, which has been introduced here for static calibration, also allows for dynamic updating using incoming information of all sorts. This is thus a unique procedure with this property. Through applying expected signatures with Bayesian model averaging, we are able to produce accurate prices without actually specifying any specific model. Hence, this method has the potential to price and hedge path-dependent options in high dimensional settings with dynamic information updating.

Bibliography

- Amrouk, E.M., Heckelei, T., 2020. Forecasting international sugar prices: A bayesian model average analysis. *Sugar Tech* 22, 552–562.
- Arribas, I.P., Salvi, C., Szpruch, L., 2020. Sig-sdes model for quantitative finance, in: *Proceedings of the First ACM International Conference on AI in Finance*, pp. 1–8.
- Bachelier, L., 1900. Théorie de la spéculation, in: *Annales scientifiques de l'École normale supérieure*, pp. 21–86.
- Black, F., Scholes, M., 1973. The Pricing of Options and Corporate Liabilities. *Journal of political economy* 81, 637–654.
- Boedihardjo, H., Geng, X., Souris, N.P., 2020. Path Developments and Tail Asymptotics of Signature for Pure Rough Paths. *Advances in Mathematics* 364, 107043.
- Breedon, D.T., Litzenberger, R.H., 1978. Prices of state-contingent claims implicit in option prices. *Journal of business* , 621–651.
- Chen, K.T., 1957. Integration of Paths, Geometric Invariants and a Generalized Baker- Hausdorff Formula. *Annals of Mathematics* 65, 163.
- Chevyrev, I., Kormilitzin, A., 2016. A Primer on the Signature Method in Machine Learning. *arXiv preprint arXiv:1603.03788* .
- Choi, J., Wu, L., 2021. A note on the option price and ‘Mass at zero in the uncorrelated SABR model and implied volatility asymptotics’. *Quantitative Finance* 21, 1083–1086.
- Cox, J.C., Ingersoll, J.E., Ross, S.A., 1985. A theory of the term structure of interest rates. *Econometrica* 53, 385–407. URL: <http://www.jstor.org/stable/1911242>.
- Cox, J.C., Ingersoll Jr, J.E., Ross, S.A., 2005. A Theory of the Term Structure of Interest Rates, in: *Theory of valuation*. World Scientific, pp. 129–164.

- Cuchiero, C., Gazzani, G., Möller, J., Svaluto-Ferro, S., 2023a. Joint calibration to spx and vix options with signature-based models. arXiv preprint arXiv:2301.13235 .
- Cuchiero, C., Gazzani, G., Svaluto-Ferro, S., 2022. Signature-based models: theory and calibration. arXiv preprint arXiv:2207.13136 .
- Cuchiero, C., Keller-Ressel, M., Teichmann, J., 2012. Polynomial processes and their applications to mathematical finance. *Finance and Stochastics* 16, 711–740.
- Cuchiero, C., Svaluto-Ferro, S., Teichmann, J., 2023b. Signature sdes from an affine and polynomial perspective. arXiv preprint arXiv:2302.01362 .
- Duembgen, M., Rogers, L., 2014. Estimate nothing. *Quantitative Finance* 14, 2065–2072.
- Duffie, D., Filipović, D., Schachermayer, W., 2003. Affine processes and applications in finance. *The Annals of Applied Probability* 13, 984–1053.
- Feller, W., 1951. Two singular diffusion problems. *Annals of mathematics* , 173–182.
- Fragoso, T.M., Bertoli, W., Louzada, F., 2018. Bayesian Model Averaging: A Systematic Review and Conceptual Classification. *International Statistical Review* 86, 1–28.
- Gyurkó, L.G., Lyons, T., 2010. Rough Paths Based Numerical Algorithms in Computational Finance. *Contemporary Mathematics* 515, 17–46.
- Gyurkó, L.G., Lyons, T., Kontkowski, M., Field, J., 2013. Extracting information from the signature of a financial data stream. arXiv preprint arXiv:1307.7244 .
- Hagan, P.S., Kumar, D., Lesniewski, A.S., Woodward, D.E., 2002. MANAGING SMILE RISK. *The Best of Wilmott* 1, 249–296.
- Heston, S.L., 1993. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The review of financial studies* 6, 327–343.
- Hinne, M., Gronau, Q.F., van den Bergh, D., Wagenmakers, E.J., 2020. A Conceptual Introduction to Bayesian Model Averaging. *Advances in Methods and Practices in Psychological Science* 3, 200–215.
- Kidger, P., Lyons, T., 2020. Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu. arXiv preprint arXiv:2001.00706 .
- Király, F.J., Oberhauser, H., 2019. Kernels for sequentially ordered data. *Journal of Machine Learning Research* 20.

- Kusuoka, S., 2004. Approximation of expectation of diffusion processes based on Lie algebra and Malliavin calculus. *Advances in mathematical economics* , 69–83.
- Levin, D., Lyons, T., Ni, H., 2013. Learning from the past, predicting the statistics for the future, learning an evolving system. arXiv preprint arXiv:1309.0260 .
- Lyons, T., Nejad, S., Arribas, I.P., 2019. Numerical method for model-free pricing of exotic derivatives using rough path signatures. arXiv preprint arXiv:1905.01720 .
- Lyons, T.J., 1998. Differential equations driven by rough signals. *Revista Matemática Iberoamericana* 14, 215–310.
- Ree, R., 1958. Lie elements and an algebra associated with shuffles. *Annals of Mathematics* 68, 210–220. URL: <http://www.jstor.org/stable/1970243>.
- Reizenstein, J., Graham, B., 2018. The iisignature library: efficient calculation of iterated-integral signatures and log signatures. arXiv preprint arXiv:1802.08252 .
- Steel, M.F., 2020. Model averaging and its use in economics. *Journal of Economic Literature* 58, 644–719.
- Sullivan, E.J., Weithers, T.M., 1991. Louis bachelier: The father of modern option pricing theory. *The Journal of Economic Education* 22, 165–171.
- Svaluto-Ferro, S., . Affine and polynomial methods for signatures URL: <https://github.com/sarasvaluto/AffPolySig>.
- Zhang, W., Yang, J., 2015. Forecasting natural gas consumption in china by bayesian model averaging. *Energy Reports* 1, 216–220.

9 Appendix

9.1 Results of Expected Signatures for Uncorrelated Brownian Motion

The simple uncorrelated Brownian motion model dynamics are

$$dt = dt, \quad dY_t^i = dB_t^i \text{ for } i = 1, 2, \quad d\langle B_t^1, B_t^2 \rangle = 0.$$

The dynamics of the model result in the following constant, linear and quadratic terms for b and a respectively as follows

$$b_1^0 = 1, b_i^k = 0 \text{ for } i, k = 1, 2, 3,$$

$$a_{ij}^0 = 0, a_{ij}^k = 0 \text{ and } a_{ij}^{kh} = 0 \text{ for } i, j, k, h = 1, 2, 3.$$

To assess the accuracy, the Mean Squared Error was computed, by comparing the analytical expected signatures against expected signatures computed using $N = 100000$ Monte Carlo simulations. The MSE was found to be a very small number. Further inspection of the accuracy involved a graphic comparison of the

Table 5: Mean Squares Error for the Analytical and Monte Carlo Expected Signatures

Mean Squared Error
9.8523253e-07

expected signatures for the analytical formula against the Monte Carlo simulations. As depicted in Figure 10, one can see that the model is accurate and performs well.

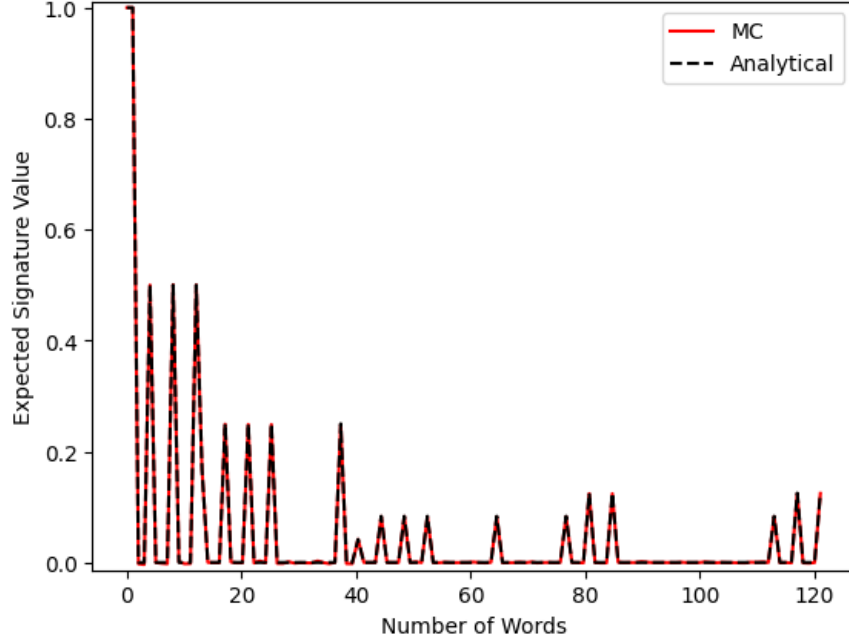


Figure 10: Sanity check for the computation of expected signatures

9.2 Bayesian Averaging in Simplified Terms

Below we shall present the use of the Bayesian averaging in simplified terms.

$$\|f\|_\infty = \lim_{p \rightarrow \infty} \|f\|_p$$

where $f : \Theta \rightarrow \mathbb{R}, f \geq 0$. Let $f = e^{-L}$, where L represents the loss function. Therefore, we obtain

$$\begin{aligned} e^{-\sup_{\theta \in \Theta} L(\theta)} &= e^{-e^{-\inf_{\theta \in \Theta} L(\theta)}} \\ &= \lim_{p \rightarrow \infty} \left(\int_{\Theta} e^{-pL(\theta)} \nu(\theta) \right)^{\frac{1}{p}} \end{aligned}$$

$$e^{-\inf_{\theta \in \Theta} L(\theta)} = - \lim_{p \rightarrow \infty} \frac{1}{p} \log \left(\int_{\Theta} e^{-pL(\theta)} \nu(\theta) \right)$$

The following theorem is made use of:

Theorem:

$$\frac{e^{-pL(\theta)} \nu(\theta)}{\int_{\Theta} e^{-pL(\theta)} \nu(\theta)} \xrightarrow{p \rightarrow \infty} \delta_{\arg \inf L}$$

9.3 Black-Scholes Formula

If the underlying follows a geometric Brownian motion, the Black-Scholes (1973) formula implies that the price of a call option and put option (respectively) at strike K is given by

$$\begin{aligned} C_t &= S_t \Phi(d_1) - e^{-r\tau} K \Phi(d_1 - \sigma\sqrt{\tau}) \\ P_t &= e^{-r\tau} K \Phi(-d_1 + \sigma\sqrt{\tau}) - S_t \Phi(-d_1) \end{aligned}$$

where, $d_1 = \frac{\log \frac{S_t}{K} + (r - q + \frac{\sigma^2}{2})\tau}{\sigma\sqrt{\tau}}$

9.4 Feller Condition

Taking into account the dynamics of the Bachelier-Heston type model

$$\begin{aligned} dS_t &= \sqrt{V_t} dW_t, \\ dV_t &= \alpha \sqrt{V_t} dB_t + \kappa(\theta - V_t)dt, \end{aligned}$$

one needs to ensure that $V_t \geq 0$. By considering the boundary classification criteria, it shows that V_t can reach zero if $\alpha^2 > 2\kappa\theta$ Cox et al. (2005). With $\kappa, \theta \geq 0$, this type of model corresponds to a continuous time first-order autoregressive process. Thus, to ensure inaccessibility of the origin in order to preclude negative volatilities (Cox et al., 2005), κ, θ and α are chosen such that

$$2\kappa\theta \geq \alpha^2.$$

This fundamental solution was derived by Feller (1951). See (Feller, 1951, Lemma 8).

9.5 Breeden-Litzenberger formula for variances swaps in Bachelier-type models

Breeden and Litzenberger (1978) proposed a method to derive the distribution for an underlying asset from extractable option prices. The Breeden Litzenberger (1978) approach can also be applied to the Bachelier-Heston type model. Since

$$S_T^2 = S_0^2 + \int_0^T 2S_t \sqrt{V_t} dW_t + \int_0^T V_t dt,$$

the variance swap with maturity T , given by $\mathbb{E}[\int_0^T V_s ds]$ can be replicated as

$$\begin{aligned} \mathbb{E}\left[\int_0^T V_s ds\right] &= \mathbb{E}[S_T^2] - S_0^2 \\ &= \int_{0 \leq K < \infty} 2\mathbb{E}[(S_T - K)^+] dK - S_0^2. \end{aligned}$$

To obtain market prices for variance swaps, we set $S_0 = 1$ and replace $\mathbb{E}[(S_T - K)^+]$ with the market call prices derived using the Black-Scholes formula.

Physics-informed Neural Networks for Option Pricing and Hedging

TEAM 2

KATE DAVIES, University of Cape Town
AMOGELANG MOTLOUTSI, University of Cape Town
NTSIKA MTHWAZI, University of Cape Town
HANNA WUTTE, ETH Zürich

Supervisor:
RALPH RUDD, University of Reykjavik

African Institute of Financial Markets and Risk Management

Contents

1	Introduction	2
1	Overview	2
1.1	Quadrature techniques	3
1.2	Deep learning and PINNs	4
2	Motivation	6
2	Methodology	7
1	Usings PINNs to approximate PDE solutions	7
1.1	Neural networks	7
1.2	Physics-informed neural networks	8
2	Option pricing via the Black-Scholes PDE	10
3	Option pricing via the transition probability density	11
3.1	Estimating transition densities	12
3.1.1	The forward Kolmogorov equation for the transition density	13
3.1.2	The backward Kolmogorov equation for the cumulative distribution function	13
3.2	Estimating integrals using quadrature methods	15
3.2.1	The Gauss-Kronrod method	15
3	Experimental analysis	16
1	Experimental setup	16
2	Option pricing via the Black-Scholes PDE	17
2.1	Pricing	17
2.2	Greeks	19
3	Option pricing via the TPDF	22

3.1	Call option	25
3.1.1	Pricing	25
3.1.2	Greeks	26
3.2	Put option	30
4	Summary	31
4	Conclusions	33

Chapter 1

Introduction

1 Overview

The pricing and hedging of derivative securities are important and common tasks in day-to-day finance. A derivative security is a financial instrument whose value depends on an underlying asset or group of assets. The most common derivative contracts are written on equities, interest rates, bonds, currencies, market indices, and commodities. Banks, corporates, governments, and risk managers all around the world rely on these instruments to hedge against risk as well as to take advantage of potential profit opportunities.

Options are some of the most commonly traded derivative instruments. Option pricing theory is used in the valuation of these derivatives. Option pricing theory is a probabilistic approach to valuing derivative contracts. Under this theory, the option's price is the expected value of its discounted payoff under the risk-neutral measure. In continuous time, this can be written as an integration problem where the integrand is a product of the payoff function and the transition probability density function (TPDF) of the underlying. Depending on the model of the underlying, the analytical expression for the TPDF may not be straightforward to obtain. Therefore, another line of research based on data-driven models has been established in order to efficiently price financial derivatives in a way that is both quick and accurate.

Hutchinson et al. (1994) demonstrated that when it comes to approximating the op-

tion pricing function, a neural network (NN) outperforms the Black (1973) model in terms of both precision and computational efficiency. Yang and Lee (2011), Liu et al. (2019) and many others agree that an artificial neural network is particularly effective at valuing options. Our report is on pricing options using physics-informed neural networks (PINNs). PINNs are neural networks trained to approximate solutions of partial differential equations (PDEs). For option pricing, PINNs can be used in several ways. Initially, we determine an option's value via PINNs approximations of solutions to the corresponding Black-Scholes PDE. Subsequently, we adopt the methodology proposed by Su et al. (2021) to price options. This involves employing a PINN to approximate TPDFs and using quadrature methods for pricing.

1.1 Quadrature techniques

Quadrature methods are techniques used to approximate the definite integral of a function. These methods approximate the area under the function curve by dividing it into smaller regions and summing up the contributions from each section.

The idea behind quadrature methods is to replace the integrand with a simpler function that can be easily integrated. This is typically done by approximating the function as a polynomial or using a set of predetermined points and weights. The Newton-Cotes formulae, see Abramowitz and Stegun (1968), and Gaussian quadrature, see Davis and Rabinowitz (1956) are two of the many available quadrature procedures. Newton-Cotes formulae offer simplicity and are generally suitable for smooth functions, but they may have limitations when dealing with functions that exhibit rapid variations or singularities. Sermutlu (2005) found that Gaussian quadratures provide higher accuracy than Newton-Cotes and are more robust for various types of functions, including those with challenging characteristics. However, Gaussian quadratures may involve more computational effort and time because of specific point selection and the weighting process. The choice between the two methods depends on the desired accuracy, the characteristics of the function being integrated, and the computational resources available.

This report uses the Gauss-Kronrod Laurie (1997) method because of the high accuracy it provides and its adaptive feature for integration.

Andricopoulos et al. (2003) proposed a flexible and robust method that can approximate this integral using quadrature (QUAD) technique to value derivatives. QUAD techniques are methods used to evaluate definite integrals without obtaining the analytical solution. The advantage of such methods is the fact that there is no need to find the explicit analytical solution of the integral. The methods solve the integral by approximating the integrand (transition density function in our case) to solve the integral. When considering a single underlying asset, Andricopoulos et al. (2003) found this approach to be faster than traditional derivative pricing methods such as lattice methods, Monte-Carlo, and finite difference. The method also proved to be powerful in pricing various options such as path-dependent options where the path is monitored at discrete time points.

Andricopoulos et al. (2007) extended this QUAD method to pricing options involving one or more underlying assets. These options also included complex path-dependent and American options. This extension offered flexibility and superior convergence and thus increased accuracy and speed: the more complex the option, the more complicated the QUAD method. The method still proved to be more accurate and faster than Monte-Carlo. Chen et al. (2014) further advanced the work of Andricopoulos et al. (2007) to make it universal for all underlying processes. In all these three papers, the authors did not address the curse of dimensionality drawback. Andricopoulos et al. (2007) noted that the limitation is the accuracy of the approximation method when the model for the underlying becomes more complex (e.g., the Heston model Heston (1993)). The QUAD technique approximates the transition probability density function with less accuracy and provides solutions for a single set of parameters. This limitation can be addressed by using the deep learning approach. The deep learning approach approximates the underlying density using neural networks and provides solutions for a range of parameters.

1.2 Deep learning and PINNs

Deep learning is a form of machine learning that uses artificial neural networks with multiple layers enabling computers to process and learn from massive amounts of data and make predictions (Goodfellow et al., 2016). It involves training these

neural networks to extract complex features from the input data and use them for classification, regression, and pattern recognition tasks. Deep learning has been used successively in a number of fields including computer vision, natural language processing, and audio analysis (LeCun et al., 2015). The use of deep learning to solve ordinary differential equations (ODEs) and, more importantly, partial differential equations (PDEs) has recently gained popularity under the name of physics-informed deep learning. Adopting this paradigm, we can replace conventional numerical methods with a neural network to approximate the solution to the differential equation. A key step in using deep learning to approximate PDE solutions is constraining the neural network to minimise the PDE residual. Several methods have been suggested for achieving this.

In Lee and Kang (1990) and Dissanayake and Phan-Thien (1994) one can find early versions of neural network algorithms that solve differential equations. van Milligen et al. (1995), Lagaris et al. (1998) and Lagaris et al. (1998) develop similar neural network strategies. Yadav et al. (2015) and Schmidhuber (2015) define and review a variety of neural network architectures for solving differential equations that are found in the literature. By adopting a mesh-free approach, deep learning overcomes the drawback of traditional numerical methods like the finite difference method (FDM) and finite element method (FEM), which require data sampling using a mesh. The mesh-free approach is centered on taking advantage of the automatic differentiation Raissi et al. (2019), which in turn breaks the curse of dimensionality when approximating high dimensional problems Grohs et al. (2018).

Some of these methods are restricted to solving particular types of PDEs such as elliptic and parabolic PDEs (Beck et al., 2019; Khoo et al., 2021). Other researchers minimise the corresponding energy functional by using the variational form of PDEs (Yu et al., 2018). Galerkin-type projections have also been considered because not all PDEs can be derived from a known functional (Meade Jr and Fernandez, 1994). In particular, Su et al. (2021) modified a deep learning algorithm developed by Sirignano and Spiliopoulos called the Deep Galerkin Method (DGM) for computing solutions of Kolmogorov PDEs. We follow that same approach by employing physics-informed neural networks (PINNs) introduced in Raissi et al. (2019) for option pricing via the transition density. To our knowledge, Su et al. (2021) are the first to price options by approximating the transition probability density using

deep learning. However, they potentially used an unnecessarily complex architecture for their neural network compared to the one that we use in this report.

2 Motivation

The primary objective of this report is to utilize PINNs and capitalize on their well-established capabilities as universal function approximators to determine option prices. In this respect, we analyze the use of PINNs in two ways. First, for approximating solutions to Black-Scholes PDEs (approach 1), and second for approximating solutions to Kolmogorov PDEs (approach 2). By extending these PDEs to parametric PDEs, we can price options for various parameter ranges without the need to retrain the neural network. This scalability is a crucial aspect of deep learning in solving parametric problems, specifically in option pricing. Additionally, it proves beneficial for risk management in the finance industry as we seek to compute sensitivities and determine option prices and densities for the underlying asset, considering factors like interest rates and volatility, in addition to spot price and time.

In a core part of this work, we employ PINNs to approximate the cumulative distribution function (CDF), i.e., the solution to a Kolmogorov PDE. With automatic differentiation embedded within PINNs, we then obtain the TPDF by taking the derivative of the CDF. To price options on the underlying asset, we employ a numerical quadrature technique called the Gauss-Kronrod method to evaluate the integral. The integrand consists of the product of the payoff function and the transition probability density function.

The remainder of this report is structured as follows; Section 1 of Chapter 2 gives a description of neural networks and the PINN methodology. Section 2 of Chapter 2 discusses approach 1, which is using PINNs to price options directly from the Black-Scholes PDE. Section 3 of Chapter 2 discusses approach 2, which is using PINNs to approximate the TPDF and using it along with a numerical quadrature method to find the arbitrage-free price of the option. Section 1 of Chapter 3 then follows with the specifications of our neural network. Sections 2 and 3 of Chapter 3 show the results of approaches 1 and 2 respectively. Section 4 of Chapter 3 compares the experimental results and in Chapter 4, we conclude.

Chapter 2

Methodology

1 Usings PINNs to approximate PDE solutions

In this section, we introduce PINNs. We start by giving a description of neural networks which form the basis of PINNs.

1.1 Neural networks

Following the derivation of Mishra and Molinaro (2022), we give an overview of neural networks. Consider a domain $\mathbb{D} = \{(s, x) | s \in [0, T], x \in \mathbb{R}^{d_1}\}$ for some $T > 0, d_1 \in \mathbb{N}$. Let $X \in \mathbb{D}$ be an input, then X is transformed into an output through a feed-forward neural network f_θ using a layer of units called neurons. These layers are composed successively by affine-linear maps A between neurons and scalar activation functions σ within the neurons resulting in the output:

$$f_\theta(X) = A_K \circ \sigma_{k-1} \circ A_{K-1} \dots \circ \sigma_2 \circ A_2 \circ \sigma_1 \circ A_1(X), \quad (2.1)$$

where $\circ$ represents the composition of functions and σ is a univariate activation function which is nonlinear. There are multiple different choices for the activation function σ in Eqn. 2.1 (Han et al., 2017). The sigmoid activation function has been historically favoured and is expressed as

$$\sigma(z) = \frac{1}{1 + e^{-z}}, z \in \mathbb{R}. \quad (2.2)$$

However, in modern neural networks, the rectified linear unit (ReLU) is quite common

$$\sigma(z) = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{otherwise} \end{cases}. \quad (2.3)$$

Let $1 < K \in \mathbb{N}$ and define

$$A_K y_k = W_k y_k + c_k \quad \text{for } W_k \in \mathbb{R}^{d_{k+1} \times d_k}, y_k \in \mathbb{R}^{d_k}, c_k \in \mathbb{R}^{d_{k+1}}, \quad (2.4)$$

for any $1 \leq k \leq K$. Then our neural network consists of $K - 1$ hidden layers, an input layer, and an output layer. An input vector y_k is provided to the k th hidden layer consisting of d_k neurons transforming it using an affine linear map A_k and then by an activation function σ . Our network contains $d_1 + d_K + \sum_{k=1}^{K-1} d_k$ neurons. The tunable weights for our network that are to be concatenated are represented by

$$\theta = \{W_k, c_k\}, \quad \forall 1 \leq k \leq K. \quad (2.5)$$

1.2 Physics-informed neural networks

In the following, we give a formal description of PINNs and how they are used to solve PDE problems. Raissi et al. (2019) defines PINNs as neural networks that have been trained to tackle supervised learning tasks while adhering to physics principles specified by general nonlinear partial differential equations. They can be used to address two types of problems of partial differential equations: data-driven solutions and data-driven discoveries. A formulation for both problems is given in (Raissi et al., 2019).

The problems that we are solving fall into the category of data-driven PDE solutions. What we mean by data-driven solutions is that we use input-output data pairs from the physical system to train the neural network. That is, we compute the hidden state of a system given boundary data. Let $\Omega \subset \mathbb{R}^d$ be the domain with the boundary $\partial\Omega$ and consider a general nonlinear PDE :

$$\frac{\partial}{\partial s} u(s, x) + \mathcal{N}[u(s, x)] = 0, \quad x \in \Omega, \quad s \in [0, T], \quad (2.6)$$

with a terminal condition

$$u(T, x) = h(x), \quad x \in \Omega, \quad (2.7)$$

and a boundary condition

$$u(s, x) = g(s, x), \quad x \in \partial\Omega, \quad s \in [0, T], \quad (2.8)$$

where $u(s, x)$ is the hidden solution of the PDE and $\mathcal{N}[u(s, x)]$ denotes a nonlinear operator. Following the work of Raissi et al. (2019), we proceed to approximate the solution to the PDE problem using deep learning. That is, we train the physics-informed neural network $f_\theta(s, x)$ to approximate the solution $u(s, x)$. By defining the residual as

$$r_\theta(s, x) := \frac{\partial}{\partial s} f_\theta(s, x) + \mathcal{N}[f_\theta(s, x)], \quad (2.9)$$

the network $f_\theta(s, x)$ can be derived by using automatic differentiation and the chain rule for differentiating function compositions Baydin et al. (2018). The parameters of $u(s, x)$ and $r_\theta(s, x)$ can then be learned by minimising the loss function

$$\mathcal{L}(\theta) := \lambda_1 \mathcal{L}_r(\theta) + \lambda_2 \mathcal{L}_{u_T}(\theta) + \lambda_3 \mathcal{L}_{u_b}(\theta), \quad (2.10)$$

where $\mathcal{L}_r$ is the loss term which penalizes the residual, $\mathcal{L}_{u_T}$ is the loss term which enforces the terminal condition and $\mathcal{L}_{u_b}$ is the loss term which enforces the boundary condition. The λ_1 , λ_2 and λ_3 are tuning parameters which control the weightings of $\mathcal{L}_r$, $\mathcal{L}_{u_T}$ and $\mathcal{L}_{u_b}$ respectively. These loss terms are the mean squared errors that take the form

$$\begin{aligned} \mathcal{L}_r &= \frac{1}{N_r} \sum_{i=1}^{N_r} \left[r(s_r^i, x_r^i) \right]^2, \\ \mathcal{L}_{u_T} &= \frac{1}{N_T} \sum_{i=1}^{N_T} \left[u(T, x_T^i) - h_T^i \right]^2, \\ \mathcal{L}_{u_b} &= \frac{1}{N_b} \sum_{i=1}^{N_b} \left[u(s_b^i, x_b^i) - g_b^i \right]^2, \end{aligned} \quad (2.11)$$

where $\{(T, x_T^i), h_T^i\}_{i=1}^{N_T}$ denotes the terminal data, $\{(s_b^i, x_b^i), g_b^i\}_{i=1}^{N_b}$ denotes the boundary data and $\{(s_r^i, x_r^i), 0\}_{i=1}^{N_r}$ denotes collocation points for the residual that are

placed inside the domain Ω at random.

Constructing a neural network $f_\theta(s, x)$ for which $\mathcal{L}(\theta)$ is as close to zero is the ultimate goal. That is,

$$\text{find } \theta^* : \theta^* = \arg \min_{\theta} \mathcal{L}(\theta).$$

So that,

$$\nabla_{\theta} \mathcal{L}(\theta^*) = \lambda_1 \nabla_{\theta} \mathcal{L}_r(\theta^*) + \lambda_2 \nabla_{\theta} \mathcal{L}_{u_T}(\theta^*) + \lambda_3 \nabla_{\theta} \mathcal{L}_{u_b}(\theta^*) < \epsilon, \quad (2.12)$$

for a small threshold ϵ . Typically, $\mathcal{L}(\theta)$ is minimized using a stochastic gradient descent algorithm.

2 Option pricing via the Black-Scholes PDE

Under the Black and Scholes (1973) model, the price evolution of a European call option is governed by the Black-Scholes PDE

$$\frac{\partial}{\partial t} V(t, S) + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} V(t, S) + rS \frac{\partial}{\partial S} V(t, S) - rV(t, S) = 0, t \in [0, T], \quad (2.13)$$

subject to the terminal condition

$$V(T, S) = \Phi(S), \quad (2.14)$$

where V is the price of the option, S is the stock price of the underlying, r is the risk-free interest rate, and σ is a constant volatility of the stock. Eqn. 2.14 is the payoff of the option at expiry time T determined by the payoff function Φ .

We note that Eqs. 2.13-2.14 are of the form Eqs. 2.6-2.7 and hence we approximate the price of the option $V(t, S)$ using PINNs. This solution just requires the parameters t and S , but it can also be parametrically encoded as a function of σ and/or r for a given range, allowing us to obtain the price $V(t, S, \sigma, r)$ after the neural network has been trained for any values of the parameter without having to retrain the network. Since traditional numerical techniques like FDM and FEM are made to solve PDEs with fixed parameters, this method is superior to them.

This method only calculates the value of the option for a single payoff described by the terminal condition.

3 Option pricing via the transition probability density

Consider an asset with the following stochastic differential equation (SDE)

$$dS = \mu(t, S)dt + \sigma(t, S)dW_t, \quad S(0) = S_0, \quad (2.15)$$

where $(W)_{t \geq 0}$ is a standard Brownian motion. The drift and diffusion coefficients $\mu(t, S)$ and $\sigma(t, S)$ can be constant, deterministic or stochastic functions. In this report we apply the Black and Scholes (1973) framework in which the evolution of the underlying under the risk-neutral measure is

$$dS = rSdt + \sigma SdW_t, \quad S(0) = S_0. \quad (2.16)$$

By option pricing theory, the fair price of the option at time $t = 0$ on this underlying asset with the spot price S_0 and maturity time T can be expressed as

$$V(0, S_0) = \mathbb{E}^{0, S_0}[f(T, S_T)], \quad (2.17)$$

$$(2.18)$$

where the discounted payoff is

$$f(T, S_T) = e^{-rT} \Phi(S_T), \quad (2.19)$$

under the risk-neutral measure for a constant risk-free rate r and a payoff function Φ . This expectation is conditional on information at time 0 and the asset price S_0 at that time. Eqn. 2.17 can be written as

$$V(0, S_0) = \int_0^\infty f(S_T) p(0, S_0; T, S_T) dS_T, \quad (2.20)$$

with the corresponding transition probability density function (TPDF) p . The integral starts from zero, because the distribution of the underlying is log-normal in the Black and Scholes (1973) framework and the log-normal distribution is only defined for positive values. When approximating TPDFs with neural networks, it is preferable to work in the log-asset price space rather than the asset price space since this change in variables enables us to shift the input domain to regions around

the origin. Generally, this improves the training of neural networks.

Let $X(t) = \ln S(t)$, then we use Ito's lemma to get the log-space transformed SDE version of Eqn. 2.16 as follows:

$$dX = e^{-X(t)} \left(r - \frac{1}{2} \sigma^2 \right) dt + e^{-X(t)} \sigma dW_t, \quad (2.21)$$

with initial condition

$$X(0) = \ln S_0 = x_0, \quad (2.22)$$

and the log-asset price space version of Eqn. 2.20 is

$$V(0, x_0) = \int_{-\infty}^{\infty} f(e^y) p(0, x_0; T, y) dy, \quad (2.23)$$

where $X_0 = \ln S_0$ and $p(t, x; T, y)$ is the TPDF of the solution to Eqn. 2.21. This is the probability of transitioning from state x at time t to state y at time T . For any integration bounds, we use a quadrature approach to estimate integral in Eqn. 2.23. We discuss this in more detail in Section 3.2.

3.1 Estimating transition densities

The Fokker-Plank equation, also known as the Kolmogorov forward equation is a PDE that describes the evolution of the probability density function for a stochastic process. The solution to this equation is the TPDF. Its initial condition is a Dirac delta distribution, which is infinite at one point and has zero value everywhere else. Neural networks typically struggle to approximate and learn to handle such singularities (Lv et al., 2023).

The appearance of a singularity in the delta distribution makes it impractical to use PINNs for the Fokker-Plank equation. In order to remove the singularity, Huang et al. (2021) used a Gaussian distribution with a small kernel to approximate the delta distribution with a continuous probability density function. To get around this problem, we solve the backward Kolmogorov equation using the CDF and a step function as the terminal condition. Thereafter, to obtain the TPDF we differentiate the CDF utilizing automatic differentiation of PINNs.

3.1.1 The forward Kolmogorov equation for the transition density

Let X be the solution to the Eqn. 2.21 and assume that it has the transition density $p(s, x; t, y)$. Björk (2009) provides a derivation for the Kolmogorov forward equation for $p(s, x; t, y)$, it takes the form

$$\frac{\partial}{\partial s} p(s, x; t, y) = -\frac{\partial}{\partial x} \left[\left(r - \frac{\sigma^2}{2} \right) p(s, x; t, y) \right] + \frac{1}{2} \frac{\partial^2}{\partial x^2} \left[\sigma^2 p(s, x; t, y) \right], \quad (2.24)$$

$t \in [0, T]$ and $x \in \mathbb{R}$,

with the initial condition for a fixed y

$$p(0, x; 0, y) = \delta(x - y), \quad x \in \mathbb{R}, \quad (2.25)$$

where $\delta(x - y)$ is the Dirac delta distribution. The PDE problem (2.24-2.25) is with respect to the independent variables (s, x) and parameters (T, y) . The Dirac delta distribution is given as, we recall:

$$\delta(x - y) = \begin{cases} +\infty, & x = y \\ 0, & x \neq y \end{cases}, \quad x \in \mathbb{R}, \quad (2.26)$$

and $\int_{-\infty}^{+\infty} \delta(x - y) dx = 1$. The singularity at the point $x = y$ makes it a challenge to approximate the TPDF of the problem (2.24-2.25) via the deep learning approach. Thus, instead of directly solving for the TPDF we solve for the CDF first and then calculate the partial derivative of CDF with respect to the state parameter to obtain the TPDF. This method requires the backward Kolmogorov equation for the CDF.

3.1.2 The backward Kolmogorov equation for the cumulative distribution function

Let $X(T) = X_T$ be the solution of Eqn. 2.21 at the terminal time T . Then the CDF of the process X_T starting at the point (s, x) is defined as

$$\begin{aligned} C(s, x; T, y) &= \mathbb{P}(X_T \leq y | X_s = x) \\ &= \int_{-\infty}^y p(s, x; T, z) dz, \end{aligned} \quad (2.27)$$

where $\mathbb{P}(X_T \leq y | X_s = x)$ is the probability of the random variable X_T to not be greater than y given, that at time s , $X_s = x$. Björk (2009) provides the derivation for the Kolmogorov backward equation for $C(s, x; T, y)$, it takes the form

$$\frac{\partial}{\partial s} C(s, x; T, y) + \frac{\partial}{\partial x} \left[\left(r - \frac{\sigma^2}{2} \right) C(s, x; T, y) \right] + \frac{1}{2} \frac{\partial^2}{\partial x^2} \left[\sigma^2 C(s, x; T, y) \right] = 0, \quad (2.28)$$

$s \in [0, T]$ and $x \in \mathbb{R}$.

with the terminal condition for a fixed y

$$\begin{aligned} C(T, x; T, y) &= \mathbb{P}(X_T \leq y | X_T = x) \\ &= \begin{cases} 1, & x \leq y \\ 0, & x > y \end{cases}, \quad x \in \mathbb{R}. \end{aligned} \quad (2.29)$$

With respect to the negative direction of time, the PDE problem (2.28-2.29) has independent variables (s, x) and parameters (T, y) . Unlike the Fokker Planck equation 2.24, which is with respect to the positive direction of time. Since the terminal condition 2.29 is a step function that is bounded, numerically solving problem 2.28-2.29 is substantially more effective than solving problem 2.24-2.25.

Using deep learning neural networks, PINNs in our case, we can approximate the CDF and then using automatic differentiation built-in function from a deep learning library in Julia called Forward Diff, we evaluate the transition probability density by differentiating the CDF with respect to the space parameter. That is,

$$p(s, x; T, y) = \frac{\partial}{\partial y} C(s, x; T, y). \quad (2.30)$$

Numerical differentiation can be done accurately since the differentiation in Eqn. 2.30 is well defined because the CDF $C(s, x; T, y)$ is a smooth function for $t \leq T$. The solution $C(s, x; T, y)$ in problem 2.28-2.29 depends on the parameters T and y , however, we are able to add more parameters such as σ and r resulting in a CDF $C(s, x; T, y, \sigma, r)$. The advantage of this is that we can produce a universal TPDF for a range of parameters (T, y, σ, r) so that we do not need to solve the PDE again.

3.2 Estimating integrals using quadrature methods

The n -point Gaussian quadrature Davis and Rabinowitz (1956) is a method of exactly approximating definite integrals of polynomials of order $2n - 1$ or less. The method carefully selects a set of n quadrature points/nodes and corresponding weights and evaluates the integral as a weighted sum of function evaluations at these points. Formally,

$$\int_a^b f(x) dx = \sum_{i=1}^n w_i f(x_i).$$

The accuracy of the method depends on the number of points chosen. There are a number of different Gaussian quadrature methods but our method of choice for this paper is the Gauss-Kronrod Laurie (1997) method because of the high accuracy it provides and its adaptive feature for integration.

3.2.1 The Gauss-Kronrod method

This is an adaptive integration method that combines the Gaussian quadrature Davis and Rabinowitz (1956) rule with Kronrod extension to achieve higher accuracy. The method adds $n + 1$ Kronrod points to $2n - 1$ points of an n -point Gaussian quadrature Davis and Rabinowitz (1956) rule such that the resulting rule has $2n + 1$ points. It has two sets of nodes x_i and weights w_i , the Gaussian nodes and the Kronrod nodes. The Gaussian nodes and weights approximate the integral with higher accuracy and the Kronrod nodes and weights estimate the error of this approximation. If the error is larger than the tolerance, the intervals are further subdivided and the error is computed again. This process is repeated until the desired error tolerance is reached.

Chapter 3

Experimental analysis

This chapter begins by explaining the experimental setup used and then proceeds to present the results obtained for experiments run using two approaches to option pricing via PINNs. The first approach (Section 2) involves directly solving the Black-Scholes PDE for a call option while the second approach (Section 3) rather finds the TPDF for the model, and then computes the option price via a quadrature method using the payoff function and density.

1 Experimental setup

In order to obtain comparable results for the two pricing approaches, the same PINN specifications (shown in table 3.1) were used for each approach.

Table 3.1: PINN specifications

Network specification	Value
number of layers	3
number of hidden layers	2
hidden layer #1 activation function	ELU
number of nodes in hidden layer #1	16
hidden layer #2 activation function	ELU
number of nodes in hidden layer #2	16
output layer activation function	sigmoid
number of nodes in output layer	1
training algorithm	quasi random sampling
number of quasi-random points in a sample	10^5
optimisation algorithm	BFGS

The weightings of the loss function components (PDE loss and boundary condition loss) were determined using a gradient scale adaptive loss function (NeuralPDE.GradientScaleAdaptiveLoss). The loss function was minimised using the BFGS optimisation algorithm (OptimizationOptimJL.BFGS).

The domains for the training data were set as $t \in [0, 1]$, $x \in [-0.5, 0.5]$ (with $x = \ln(S) \rightarrow S \in [0.6065, 1.6487]$), $\sigma \in [0.01, 0.2]$ and $r \in [0, 0.1]$ for both approaches and a quasi-random sampling technique was used.

2 Option pricing via the Black-Scholes PDE

The Black-Scholes PDE for a call option with the boundary conditions described in Section 2 of the methodology chapter and a fixed strike of $K = 1.1$ was solved using the defined PINN to obtain a parametric function $V(t, S, \sigma, r)$ of time, asset value, volatility, and interest rate. Using this function, predictions for option prices and sensitivities were calculated for various asset values and times to maturity τ . Two error metrics were computed, namely the absolute and relative errors. The absolute error was found by taking the absolute difference between the prediction value and the analytical value and the relative error was found by dividing the absolute error by the absolute value of the analytical solution, with the modification that a term of 0.5 was added to the denominator in the fraction to resolve undefined values when $x_{\text{analytical}}$ is zero or very small. Formally,

$$\begin{aligned} \text{absolute error} &= |x_{\text{analytical}} - x_{\text{predict}}|, \\ \text{relative error (\%)} &= \frac{|x_{\text{analytical}} - x_{\text{predict}}|}{0.5 + |x_{\text{analytical}}|}. \end{aligned}$$

2.1 Pricing

Figure 3.1 shows a surface plot of the price predictions as a function of asset price divided by the strike $\frac{S}{K}$ and time to maturity τ with a fixed σ of 10% and rate r of 5%. Generally, the pricing function behaves as expected, however, relatively large errors are observed for large times to maturity τ and asset values deeper in-the-money. Errors in this region are evident in the second plot of figure 3.2 where the prediction line deviates from the analytical solution. These errors may be attributed

to the neural network not fitting the function well at the edge of the training grid, in both the asset value and time domains.

Additionally, it can be seen that for low τ values there are relatively high errors at-the-money, where $\frac{S}{K}$ is equal to 1. This is evident in 3.2 for $\tau = 0.1$. Considering the payoff function for a call option (which is specified as the boundary condition), it appears that the PINN solution is not able to capture the sharpness of the function at-the-money, where the payoff changes from a constant zero to a linearly increasing function.

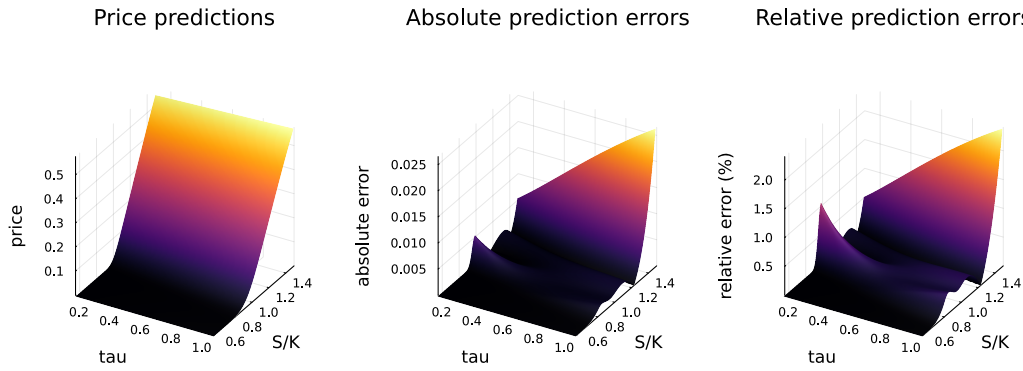


Figure 3.1: Surfaces plots of the price predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

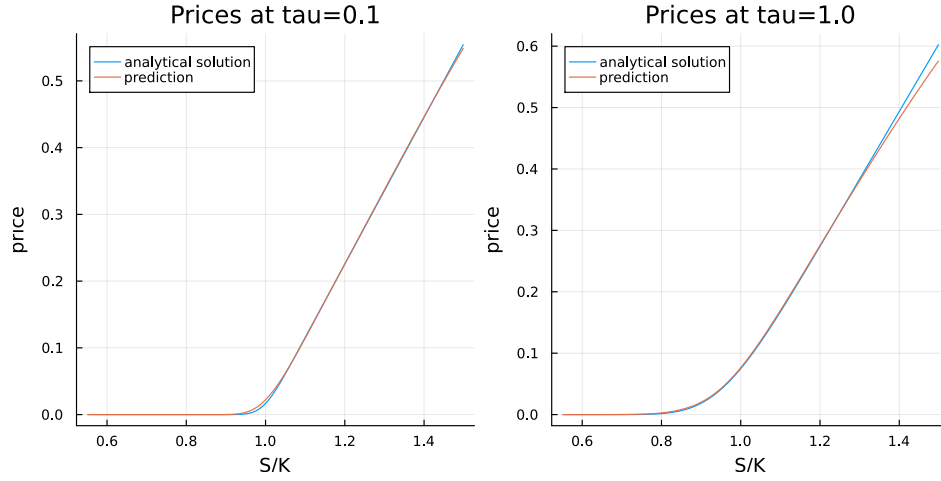


Figure 3.2: Price predictions for a call option as a function of asset price divided by strike $\frac{S}{K}$ at $\tau = 0.1$ and $\tau = 1$

2.2 Greeks

Since the call option price was found in the form of a parametric function, the Greeks were computed by simply finding the partial derivatives using an auto-differentiation method. Figures 3.3 to 3.8 show the predictions for delta ($\Delta = \frac{\partial V}{\partial S}$), vega ($\nu = \frac{\partial V}{\partial \sigma}$) and rho ($\rho = \frac{\partial V}{\partial r}$) with the fixed values again of σ at 10% and rate r at 5%.

Figures 3.3 and 3.4 for the delta values show significant errors for values deep in-the-money, which emphasises the neural network's poor fit to the upper boundary of the asset value domain. Furthermore, large relative errors are observed at lower τ values when the value is near the at-the-money point, indicating that there is a poor fit at the lower bound of the time domain and there are limitations in meeting the boundary condition specified by the call payoff.

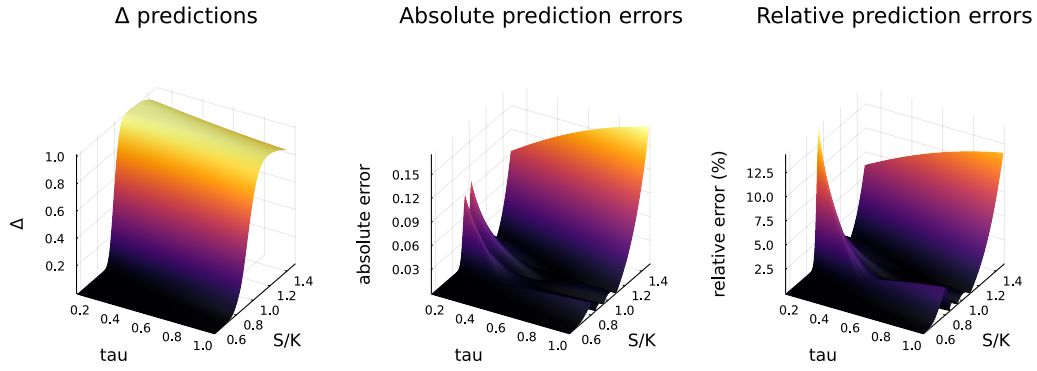


Figure 3.3: Surfaces plots of the delta predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

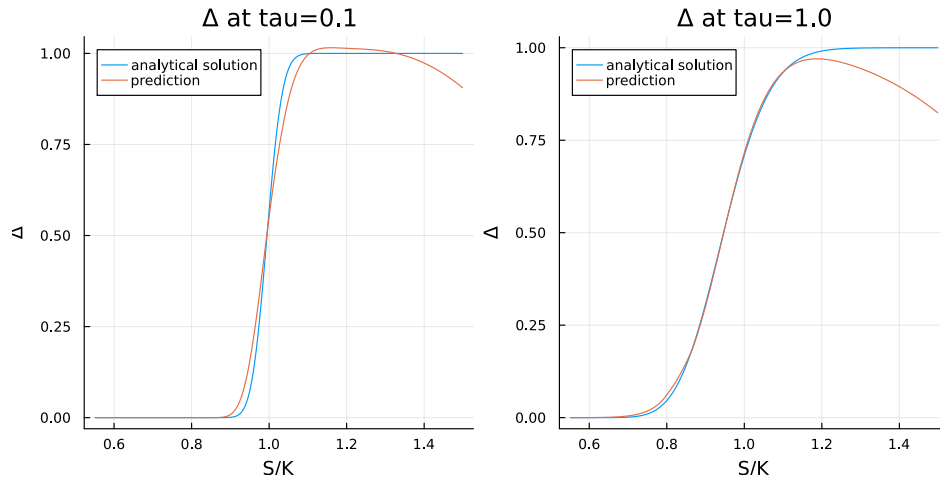


Figure 3.4: Delta predictions for a call option as a function of asset price divided by strike $\frac{S}{K}$ at $\tau = 0.1$ and $\tau = 1$

In figures 3.5 and 3.6 for the vega and figures 3.7 and 3.8 for rho, the same trends as observed for the fit of the delta are evident for both vega and rho. Most notably in figure 3.8, at $\tau = 0.1$ the prediction for rho displays unusual behaviour.

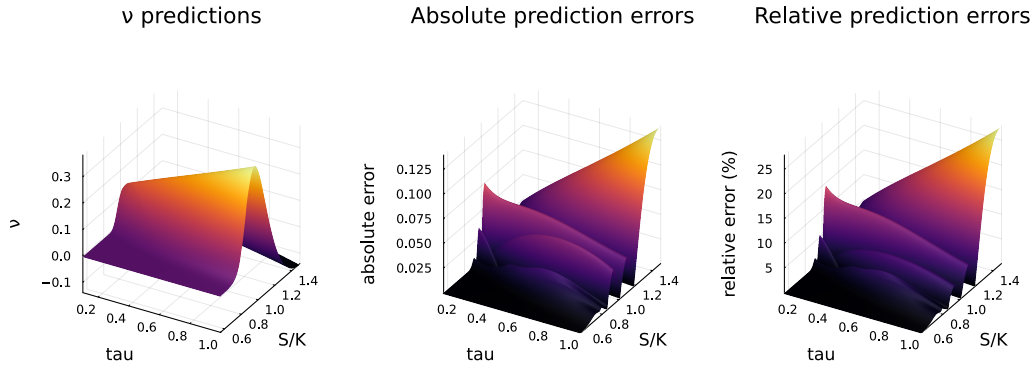


Figure 3.5: Surfaces plots of the vega predictions of a call option as a function of time to maturity (τ) and asset price divided by strike ($\frac{S}{K}$) alongside the absolute and relative errors

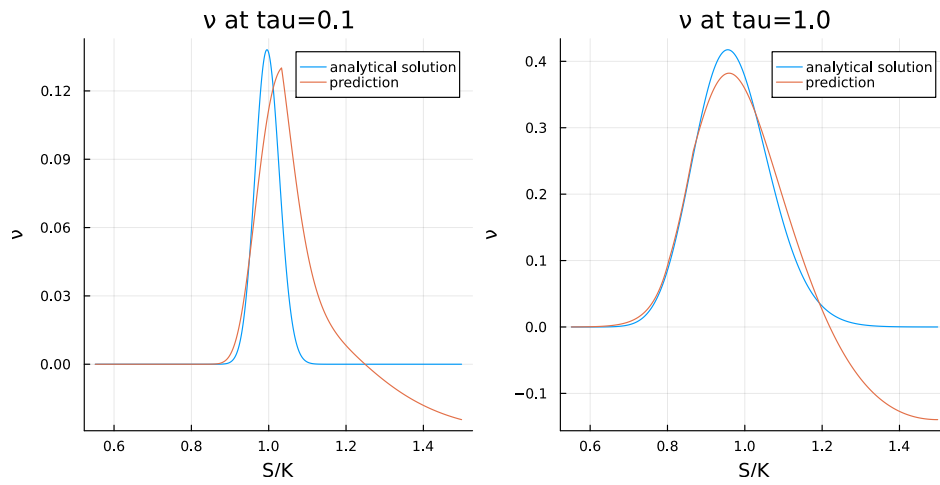


Figure 3.6: Vega predictions for a call option as a function of asset price divided by strike ($\frac{S}{K}$) at $\tau = 0.1$ and $\tau = 1$

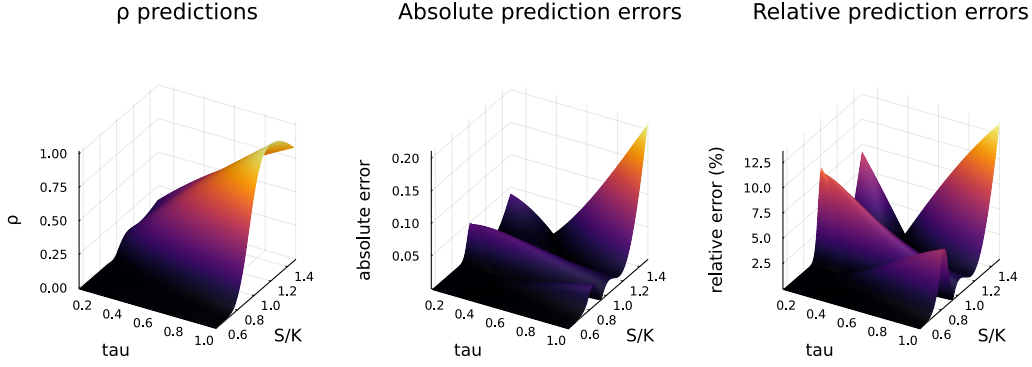


Figure 3.7: Surfaces plots of the rho predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

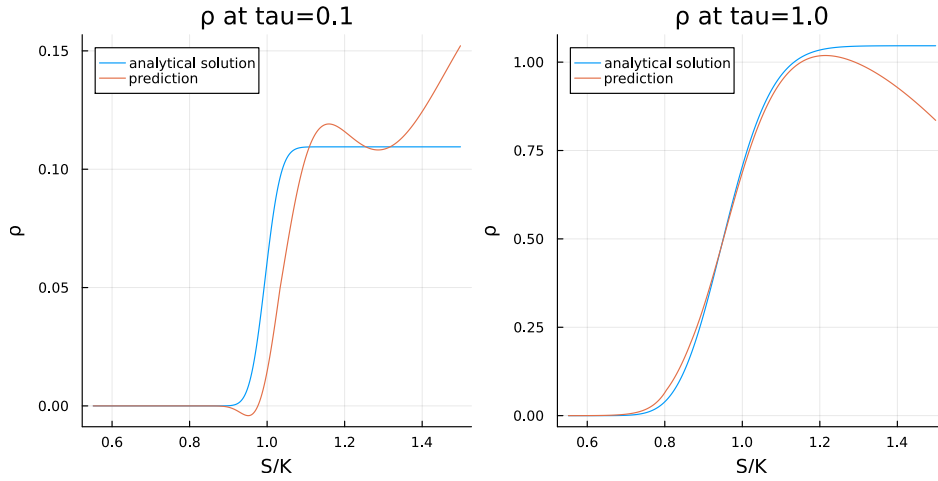


Figure 3.8: Rho predictions for a call option as a function of asset price divided by strike ($\frac{S}{K}$) at $\tau = 0.1$ and $\tau = 1$

3 Option pricing via the TPDF

Using the PINN to solve for the model density (as described in Section 3 and Equation 2.28 of the method) broadens the scope of possible applications as compared to the previous approach of option pricing. This section begins by showing the pricing and sensitivity results for a call option and comparing them to the previous method, then extends the results by pricing an additional option at almost no extra

cost.

The Greeks are computed by finding the option value as a parametric function, then using auto-differentiation to find the partial derivatives. As with the results in the previous section, σ was fixed at 10% and the rate r at 5%

Figures 3.9 to 3.12 below show the CDFs and PDFs found for the terminal asset values, which will be used to price the options specified in the following subsections. In the surface plots for the CDF, it can be seen that the largest errors arise due to the poor fit of the slopes of the predictions. The plots in figure 3.10 highlight this by showing that at both $\tau = 0.1$ and $\tau = 1$ the neural network struggles to learn the sharp steepness of the actual CDF. Although the relative error plot in figure 3.9 shows the highest error at $\tau = 0.1$, it should be noted that overall there was a better fit at $\tau = 0.1$ as evidenced by a root-mean-square error (RMSE) of about 0.08696 for this time compared to the RMSE value of about 0.1352 at $\tau = 1$. This may be owing to the boundary condition component of the loss function having a greater weighting.

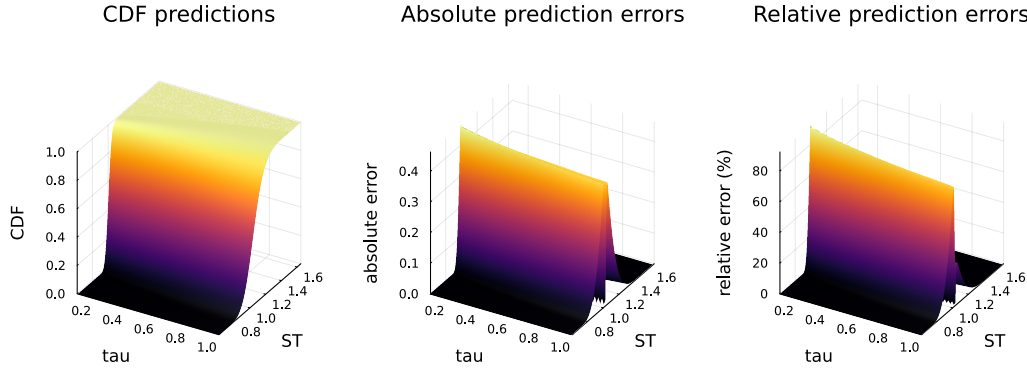


Figure 3.9: Surfaces plots of the CDF predictions as a function of time to maturity (τ) and terminal asset value alongside the absolute and relative errors

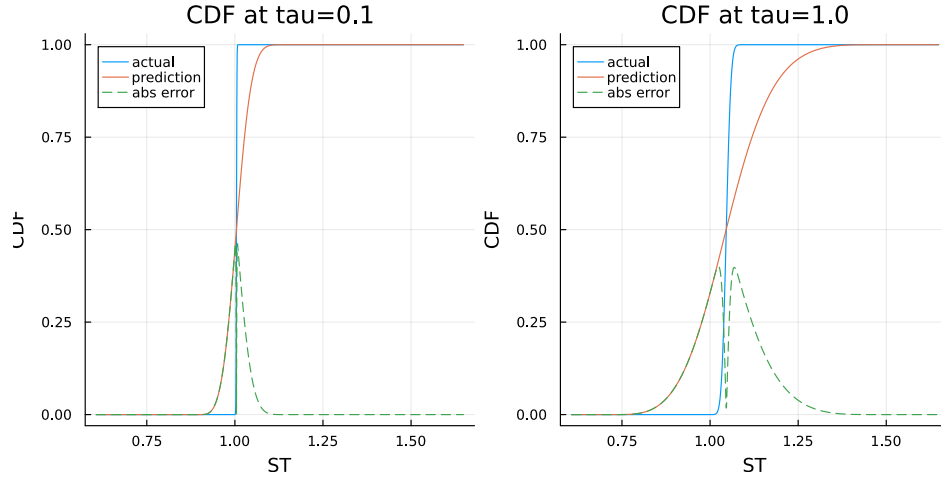


Figure 3.10: CDF predictions as a function of terminal asset value at $\tau = 0.1$ and $\tau = 1$

The PDFs (figure 3.11) were obtained from the CDF predictions using an auto-differentiation method. It is clear from the error surface plots as well as the plots in figure 3.12 that the PDF predictions have a severely poor fit to the actual solutions, which follows from the CDF predictions not capturing the steepness of the function. However, in the pricing results that follow, it will be shown that this poor fit to the density does not make this pricing approach invalid.

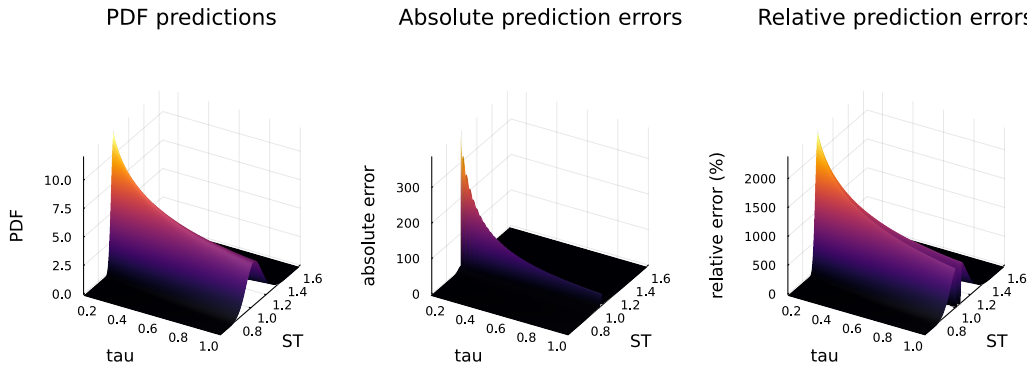


Figure 3.11: Surfaces plots of the PDF predictions as a function of time to maturity τ and terminal asset value alongside the absolute and relative errors

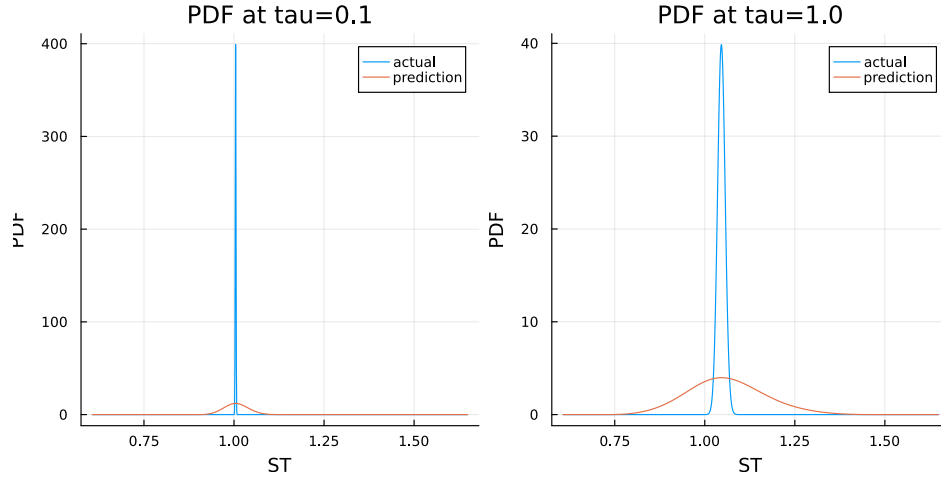


Figure 3.12: PDF predictions as a function of terminal asset value at $\tau = 0.1$ and $\tau = 1$

3.1 Call option

The value of a call option $V(t, S, \sigma, r)$ as a function of time, asset value, volatility, and interest rate was obtained using the payoff function of the option and the obtained density described above.

3.1.1 Pricing

From the absolute error surface plot for the call option price predictions in figure 3.13 it is evident that the magnitude of the errors are significantly lower than those observed for approach one. This is supported by the fact that the overall RMSE for the price predictions of the previous approach was about 0.004177, while the RMSE for the current approach is about 0.0003501 – more than ten times smaller. Although this approach boasts an improvement, the challenges of fitting the values at higher asset values and lower values of τ are still apparent. Additionally, there are also errors around the at-the-money point, as observed for approach one.

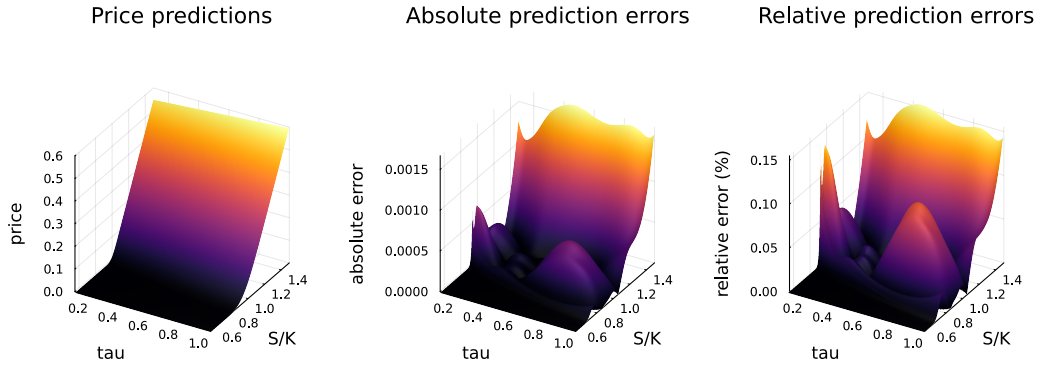


Figure 3.13: Surfaces plots of the price predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

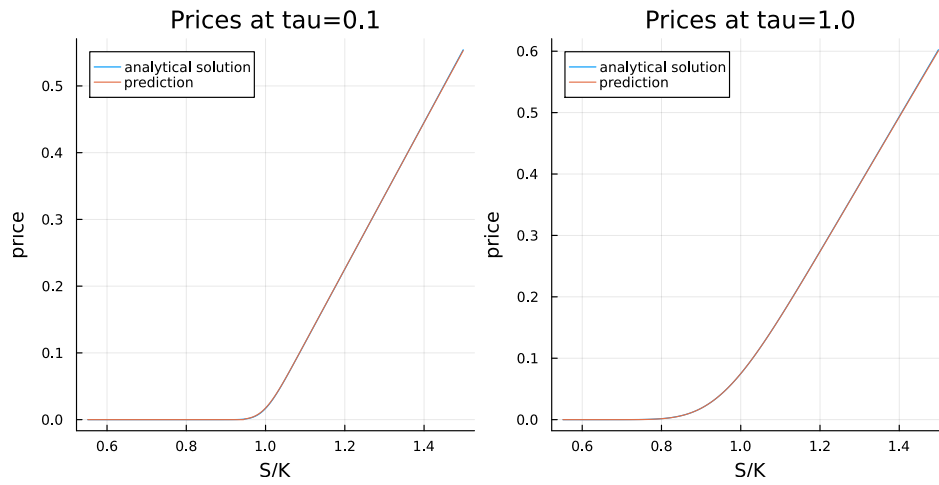


Figure 3.14: Price predictions for a call option as a function of asset price divided by strike $\frac{S}{K}$ at $\tau = 0.1$ and $\tau = 1$

3.1.2 Greeks

In addition to the improvements shown in the pricing predictions using the second approach, the computations of the Greeks delta, vega and rho also show a reduction in the overall errors as evidenced in table 3.2.

Table 3.2: Comparison of RMSEs for the delta, vega, and rho predictions of two option pricing approaches

	Delta Δ	Vega ν	Rho ρ
Approach 1 RMSE	0.04061	0.04128	0.03049
Approach 2 RMSE	0.003092	0.01873	0.01302

In the case of the delta predictions, the error plots in figure 3.15 show that the highest errors occurred at low τ values and high asset values – in line with the preceding trends observed. These errors are visible in the plots shown in figure 3.16 which demonstrates that at $\tau = 0.1$ the prediction for delta deviates from the analytical solution at both high asset values and values going out-the-money.

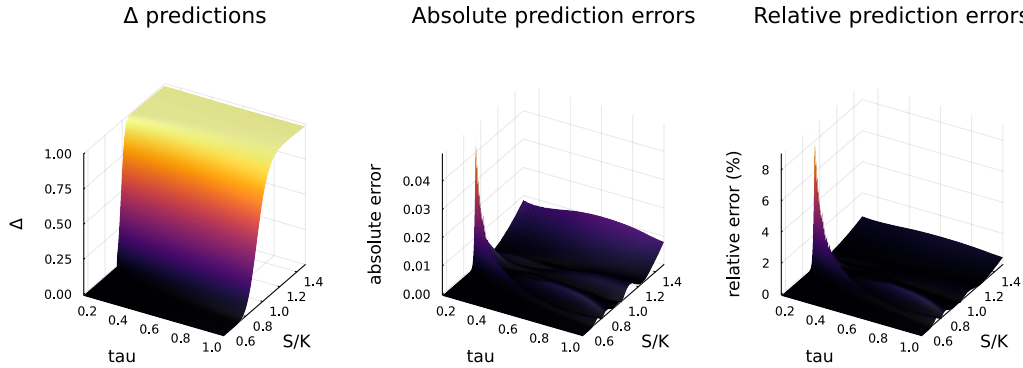


Figure 3.15: Surfaces plots of the delta predictions of a call option as a function of time to maturity (τ) and asset price divided by strike ($\frac{S}{K}$) alongside the absolute and relative errors

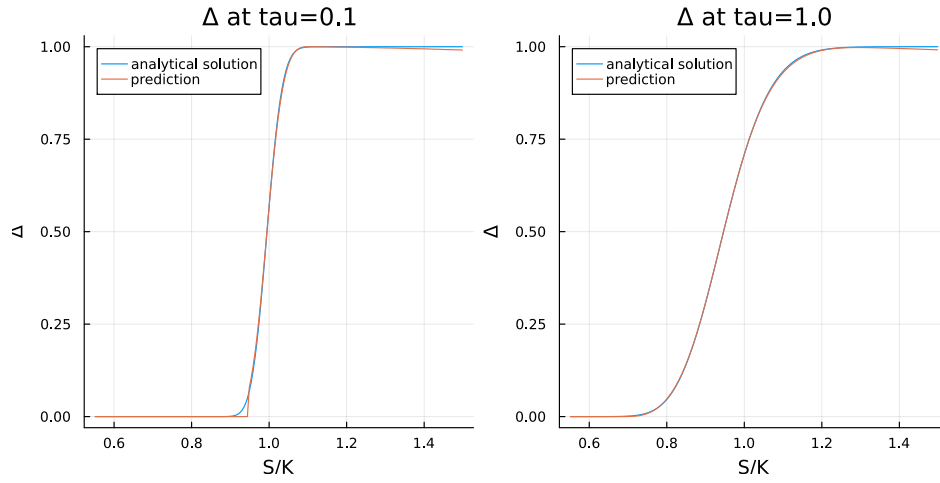


Figure 3.16: Delta predictions for a call option as a function of asset price divided by strike ($\frac{S}{K}$) at $\tau = 0.1$ and $\tau = 1$

The plots of the vega predictions in figure 3.17 show that there are relatively high errors for asset values in the money. Although the mismatch of the prediction to the analytical solution is prominent in the plots for $\tau = 0.1$ and $\tau = 1$ (figure 3.18), these results still present a significant improvement to those shown for approach one – in which case the vega predictions were negative for values deep in the money.

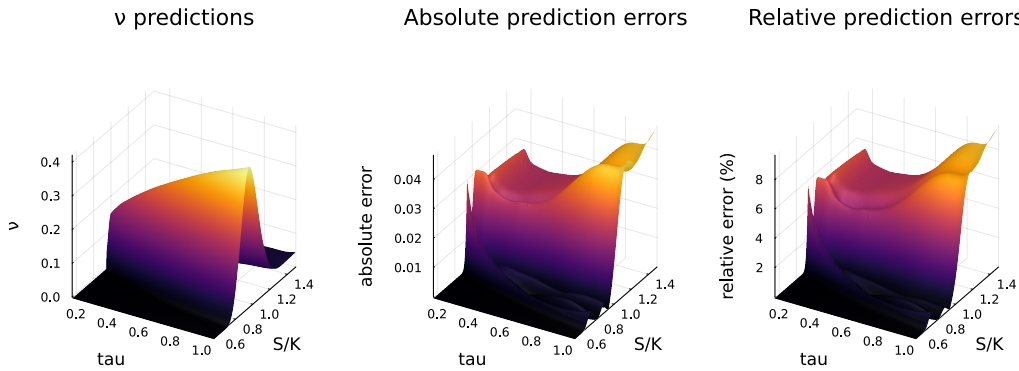


Figure 3.17: Surfaces plots of the vega predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

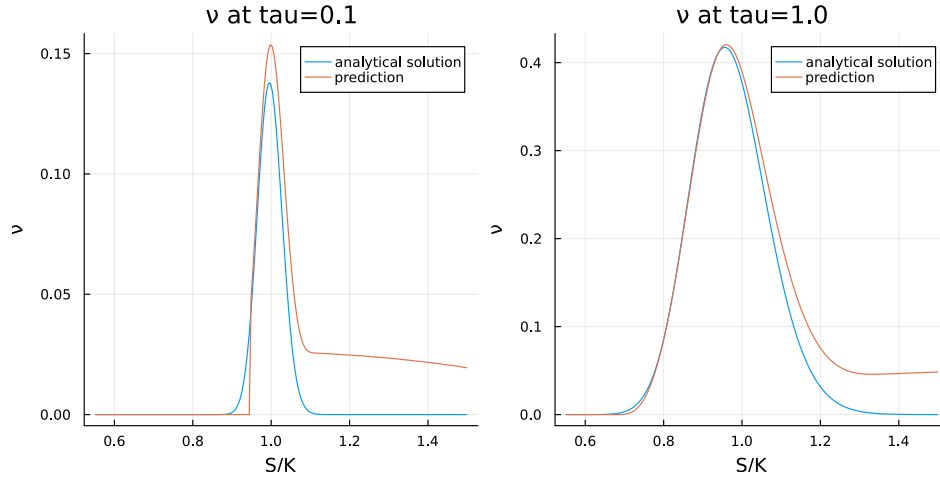


Figure 3.18: Vega predictions for a call option as a function of asset price divided by strike ($\frac{S}{K}$) at $\tau = 0.1$ and $\tau = 1$

The results for the predictions of rho (shown in figures 3.19 and 3.20) display the largest errors at the low τ value and high asset value boundaries, again indicating the difficulties for the neural network to adequately fit the edge of the training grid.

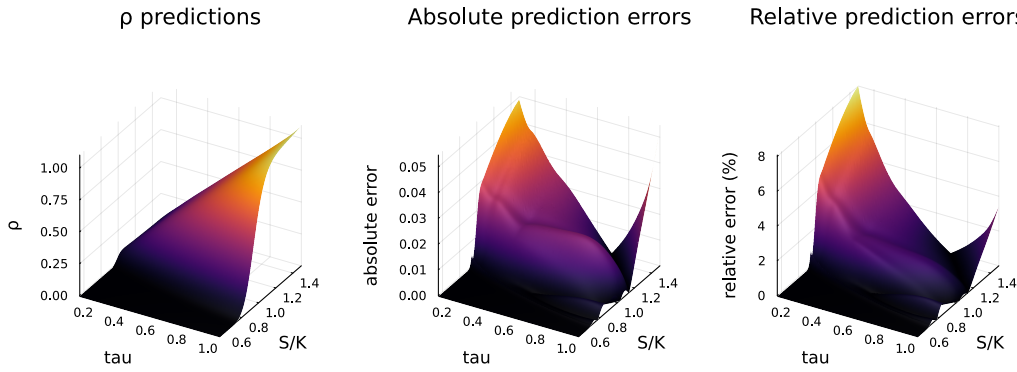


Figure 3.19: Surfaces plots of the rho predictions of a call option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

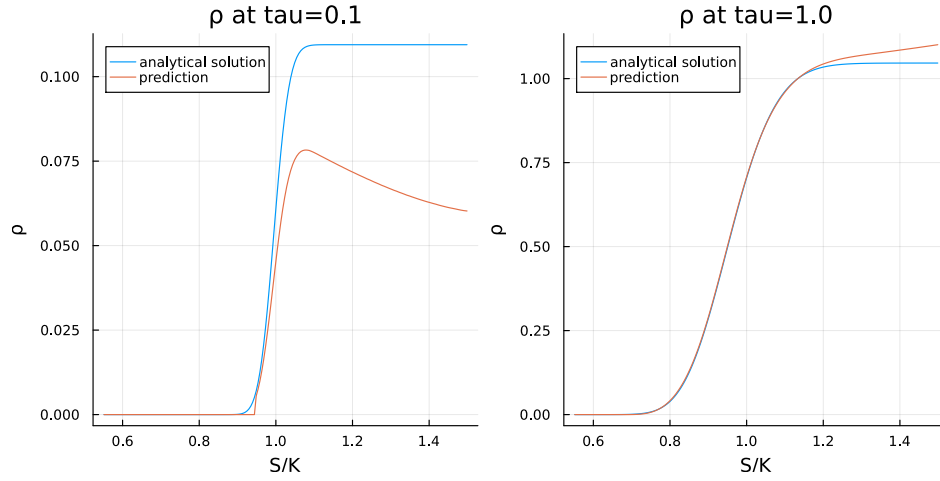


Figure 3.20: Rho predictions for a call option as a function of asset price divided by strike $\frac{S}{K}$ at $\tau = 0.1$ and $\tau = 1$

3.2 Put option

Other options that only depend on the terminal asset value can be priced using the model density in the same way as a call option by simply replacing the payoff function in the calculation. This requires significantly less computation time as compared to the first approach, which would require the PINN to be re-trained using the new boundary condition for the option payoff. The pricing results for a put option are shown in this subsection (figures 3.21 and 3.22) for demonstration. The Greeks could easily be computed following the same method as for the call option.

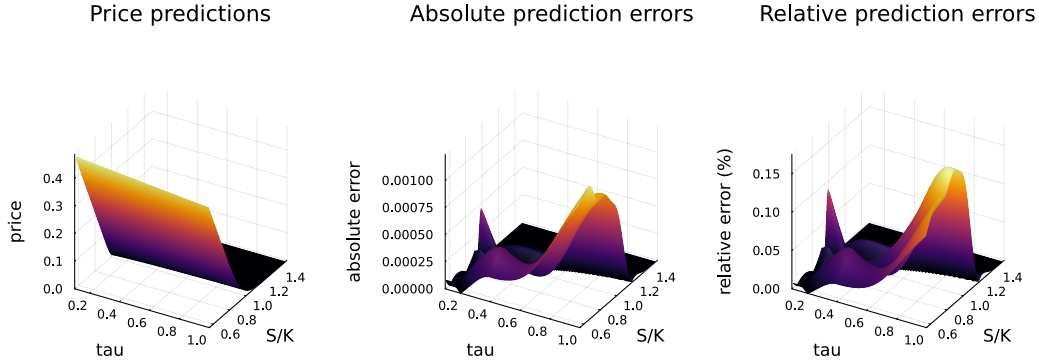


Figure 3.21: Surfaces plots of the price predictions of a put option as a function of time to maturity τ and asset price divided by strike $\frac{S}{K}$ alongside the absolute and relative errors

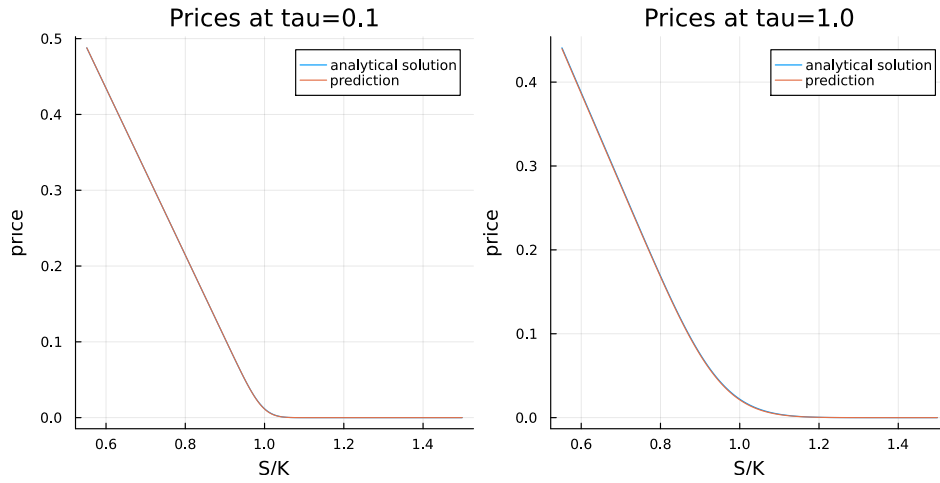


Figure 3.22: Price predictions for a put option as a function of asset price divided by strike $\frac{S}{K}$ at $\tau = 0.1$ and $\tau = 1$

4 Summary

The implementation and results of two approaches to option pricing using PINNs were investigated in this chapter to assess their feasibility and accuracy. Moreover, the experiments for each approach were run using equivalent models so that fair comparisons could be drawn between the two approaches and the advantages of each identified.

Overall, it was observed that for a call option, the second approach of using TPDFs for option pricing produced more accurate results. This was supported by lower RMSE values across all quantities computed (prices and Greeks) and lower absolute errors. It appears that even though the CDF and PDF predictions were not able to accurately capture the shape of the actual distributions, the pricing and sensitivity predictions were more accurate due to the averaging nature of the calculations involved.

In addition to the improved accuracy, the second approach also holds the advantage that other options can easily be priced using the TPDFs obtained with little extra computational cost. This is significant as the second approach offers more flexibility with a greater range of applications and this flexibility does not come at the cost of accuracy.

For both pricing approaches, the largest errors were observed at the boundaries of the training grid – particularly at low τ values and high asset values. In order to address this, a few adjustments could be made to the PINN design such as applying larger parameter domains for training then using subsets of these for calculations or using different sampling techniques to better obtain training data near the boundaries. Another factor that may affect the fit of the solution at the boundaries is the weighting of the boundary condition component of the loss function. This weighting should be carefully chosen and adjusted according to the needs of the application – for example, in the case of options whose payoffs only depend on the terminal values, it might be advantageous to apply a higher weighting to the boundary condition.

Chapter 4

Conclusions

This investigation aimed to provide a realistic view on the application of PINNs for option pricing. The relevant theory for this application was detailed and the methodology to price options using two approaches was developed. The first approach involved using PINNs to approximate option pricing PDE solutions directly while the second approach entailed approximating transition probability densities which could then be used to price a variety of options via quadrature methods. Following the development of the method, an experimental analysis was carried out to analyse the effectiveness of each approach. It was found that for a call option, under equivalent experiment conditions, the second approach performed significantly better. This was evidenced by the lower error metric scores across the quantities computed (prices and sensitivities). Although this approach produced superior results, several issues with the model were apparent – particularly limitations in fitting the boundary values of the training grid. This leads to the need to investigate more suitable design specifications for PINNs for the application of option pricing.

Indeed, general PINN training faces a number of challenges. Most prominently, differing magnitudes of the gradients of the two main loss components promoting zero PDE residuals and fitting boundary conditions, respectively, can be observed (Wang et al., 2020). Consequently, during training, it might happen that a PINN is pushed towards satisfying one loss term in expense of the other. Fortunately, the PINN framework allows for a number of design choices to improve the imbalance of loss gradients. In this investigation, we tested both manual and adaptive

weighting of the loss terms for PDE residuals and boundary conditions. In particular, we found the adaptive scheme proposed by Wang et al. (2020) to be effective in reducing overall loss.

For option pricing in particular, it might be of interest to prioritize PINN fits at certain points in the time domain. Fine-tuning the fit at these points by an increased weighting of certain loss components, therefore, is of importance. Going forward, this fine-tuning could be further addressed by targeted sampling of training data in critical regions.

Moreover, our experimental analysis showed better fits of PINNs in central regions of the domain they were trained on. Consequently, we advocate to select wider ranges for PINN training than will be used for PINN evaluation.

The main advantages of PINNs for solving PDEs are three-fold. In the course of the investigation, we tested how these advantages translate to the estimation of transition probabilities. First, the PINN framework is specifically flexible to incorporate high-dimensional input. Thus, transition probability densities for high-dimensional markets can be readily learned. Second, via parametrized PINNs, it is possible to *jointly* solve a range of parametric PDEs. For solving Kolmogorov PDEs with PINNs, this means that a single training procedure leads to estimated transition probability densities for a continuous range of asset dynamics. Third, PINNs are *gridless*. Once estimated, the transition probability densities can be readily evaluated at any space, time and parameter value of the domain.

Consequently, we found that option prices can be computed in split seconds by numeric integration. However, typical Monte Carlo simulations might be comparably quick. It is up to future research to contrast these approaches on a range of underlying dynamics.

Concluding our analysis, we believe that the true stronghold of PINNs for TPDFs lies in risk management. With the PINN technology, it is possible to pre-compute option derivatives as functions on the domain. These functions can then be readily evaluated at any time, space, or parameter input without the need of computing finite differences. In particular, we see great potential for the use of PINNs in *dynamic* risk management. Since derivatives can be computed continuously in time and space, the PINN technology allows to quickly adapt to changes in the underlying or the portfolio setup.

In this work, we presented a proof of concept of PINNs in pricing and risk manage-

ment in a Black Scholes framework. Going forward, it will be particularly exciting to investigate the full potential of the PINN technology for estimating transition probability densities under more general asset dynamics.

Bibliography

- Abramowitz, M., Stegun, I.A., 1968. Handbook of mathematical functions with formulas, graphs, and mathematical tables. volume 55. US Government printing office.
- Andricopoulos, A.D., Widdicks, M., Duck, P.W., Newton, D.P., 2003. Universal option valuation using quadrature methods. *Journal of Financial Economics* 67, 447–471.
- Andricopoulos, A.D., Widdicks, M., Newton, D.P., Duck, P.W., 2007. Extending quadrature methods to value multi-asset and complex path dependent options. *Journal of Financial Economics* 83, 471–499.
- Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research* 18, 1–43.
- Beck, C., E, W., Jentzen, A., 2019. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science* 29, 1563–1619.
- Björk, T., 2009. Arbitrage theory in continuous time. Oxford university press.
- Black, F., Scholes, M., 1973. The pricing of options and corporate liabilities. *Journal of political economy* 81, 637–654.
- Chen, D., Härkönen, H.J., Newton, D.P., 2014. Advancing the universality of quadrature methods to any underlying process for option pricing. *Journal of Financial Economics* 114, 600–612.

- Davis, P., Rabinowitz, P., 1956. Abscissas and weights for gaussian quadratures. *Journal of Research of the National Bureau of Standards* 56, 35.
- Dissanayake, M., Phan-Thien, N., 1994. Neural-network-based approximations for solving partial differential equations. *communications in Numerical Methods in Engineering* 10, 195–201.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep learning*. MIT press.
- Grohs, P., Hornung, F., Jentzen, A., Von Wurstemberger, P., 2018. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations. *arXiv preprint arXiv:1809.02362*.
- Han, J., Jentzen, A., et al., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in mathematics and statistics* 5, 349–380.
- Heston, S.L., 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies* 6, 327–343.
- Huang, X., Liu, H., Shi, B., Wang, Z., Yang, K., Li, Y., Weng, B., Wang, M., Chu, H., Zhou, J., et al., 2021. Solving partial differential equations with point source based on physics-informed neural networks. *arXiv preprint arXiv:2111.01394*.
- Hutchinson, J.M., Lo, A.W., Poggio, T., 1994. A nonparametric approach to pricing and hedging derivative securities via learning networks. *The journal of Finance* 49, 851–889.
- Khoo, Y., Lu, J., Ying, L., 2021. Solving parametric pde problems with artificial neural networks. *European Journal of Applied Mathematics* 32, 421–435.
- Lagaris, I.E., Likas, A., Fotiadis, D.I., 1998. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks* 9, 987–1000.
- Laurie, D., 1997. Calculation of gauss-kronrod quadrature rules. *Mathematics of Computation* 66, 1133–1145.

- LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature* 521, 436–44. doi:10.1038/nature14539.
- Lee, H., Kang, I.S., 1990. Neural algorithm for solving differential equations. *Journal of Computational Physics* 91, 110–131.
- Liu, S., Oosterlee, C.W., Bohte, S.M., 2019. Pricing options and computing implied volatilities using neural networks. *Risks* 7, 16.
- Lv, C., Wang, L., Xie, C., 2023. A hybrid physics-informed neural network for nonlinear partial differential equation. *International Journal of Modern Physics C* 34, 2350082.
- Meade Jr, A.J., Fernandez, A.A., 1994. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling* 19, 1–25.
- van Milligen, B.P., Tribaldos, V., Jiménez, J., 1995. Neural network differential equation and plasma equilibrium solver. *Physical review letters* 75, 3594.
- Mishra, S., Molinaro, R., 2022. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for pdes. *IMA Journal of Numerical Analysis* 42, 981–1022.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378, 686–707.
- Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks* 61, 85–117.
- Sermutlu, E., 2005. Comparison of newton–cotes and gaussian methods of quadrature. *Applied mathematics and computation* 171, 1048–1057.
- Su, H., Tretyakov, M.V., Newton, D.P., 2021. Option valuation through deep learning of transition probability density. *arXiv preprint arXiv:2105.10467*.
- Wang, S., Teng, Y., Perdikaris, P., 2020. Understanding and mitigating gradient pathologies in physics-informed neural networks. *CoRR abs/2001.04536*. URL: <https://arxiv.org/abs/2001.04536>, arXiv:2001.04536.

- Yadav, N., Yadav, A., Kumar, M., et al., 2015. An introduction to neural network methods for differential equations. volume 1. Springer.
- Yang, S.H., Lee, J., 2011. Predicting a distribution of implied volatilities for option pricing. *Expert Systems with Applications* 38, 1702–1708.
- Yu, B., et al., 2018. The deep ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics* 6, 1–12.

Sequential Monte Carlo for Index Tracking with Transaction Costs

TEAM 3

LEILA HAMILTON-RUSSELL, University of Cape Town
THOMAS O'CALLAGHAN, University of Cape Town
FRANCESCA PRIMAVERA, University of Vienna
DMITRII SAVIN, University College London

Supervisor:

ERIK SCHLOGL, University of Technology Sydney

African Institute of Financial Markets and Risk Management

Contents

1	Introduction	3
2	Preliminaries	5
2.1	The Index Tracking Problem	5
2.1.1	Optimization problem as sampling problem	5
2.2	Sequential Monte Carlo Methods	6
2.2.1	Sequential Importance Sampling	7
2.2.2	Resampling	9
2.2.3	Support Boosting and Metropolis Hastings Algorithm	9
3	Transaction Costs	11
3.0.1	Dynamics of the wealth process given proportional transaction costs	11
3.0.2	Index tracking problem with proportional transaction costs	12
4	The SMC Algorithm with Principal Component Analysis	14
5	Methodology and Data	15
5.1	Models	15
5.2	Constraints and Optimisation	16
5.3	Data	17
5.4	Performance Metrics	19
6	Results	21
7	Conclusions	26

1 Introduction

Index tracking, also known as index replication, is a passive investment strategy that has gained popularity due to the higher costs associated with active investing and skepticism regarding the ability of active funds to outperform the market (see Rompotis (2013), Barber and Odean (2000)). Financial indices track the performance of a collection of financial assets or an overall market. Indices are not tradable instruments themselves, however replication can be achieved by creating a portfolio of assets that closely match the performance of an index. To perfectly replicate the performance of an index, one would need to invest in all its constituents stocks in the appropriate proportions. This may require continuously trading in hundreds of assets, which is both costly and often infeasible. More specifically, index replication's effectiveness can be influenced by the constituent assets' liquidity. Furthermore, continuous trading can lead to high transaction costs due to the costs incurred with each trade and the adjustments made during rebalancing of the portfolio's composition. The goal of sparse index tracking is create a portfolio that closely replicates the performance of a benchmark index while avoiding the need to invest in all its constituent assets. The desired solution is a portfolio of a relatively small subset of assets that replicates an index at a lower cost.

Satpathy and Shah (2022) proposed a Sequential Monte Carlo (SMC) algorithm to address the sparse index tracking problem. The key idea is to interpret the index-tracking problem as a maximization problem over the set of all possible (fixed-size) permutations of stocks composing some index and then using SMC as a global optimization technique. Despite the novelty and the success of this approach when compared to other methods, Tibshirani (1996), Xu et al. (2015), Benidis et al. (2018), it is not clear whether this technique is an optimal index-tracking strategy as it falls short in addressing the equally significant objective of reducing transaction costs. Indeed, when rebalancing the tracking portfolio, the algorithm does not explicitly minimize transaction costs. It focuses solely on minimising the tracking error between the portfolio and index returns. This approach may lead to marginal improvements in the tracking error, but the cost associated with the portfolio rebalancing could outweigh any benefit gained.

The purpose of this work is to improve the Sequential Monte Carlo algorithm proposed by Satpathy and Shah (2022) by accounting for transaction costs in portfolio rebalancing. To this end, we redefine the dynamics of the index-tracking portfolio in such a way as to account for (proportional) transaction costs. Relying on this, we first reformulate the problem by adding a penalization term to the objective function of the index-tracking problem and then study this modified problem by adopting similar techniques.

The remainder of the report is structured as follows. The general formulation of the index tracking problem, as proposed by Satpathy and Shah (2022), is presented in Section 2. In Section 3, a description of how to adjust the index tracking prob-

lem for transaction costs is given. Section 5 details the methods used for model implementation and testing, followed by the results in Section 6. Finally, the report concludes in Section 7 with a brief summary of the outcomes as well as a short description of possible further studies.

2 Preliminaries

2.1 The Index Tracking Problem

In this section, we recall the general formulation of the index-tracking problem.

Let us consider an index I comprised of N stocks. Denote by $\mathbf{r}_I \in \mathbb{R}^T$ the column vector of daily return of the index over a period of T days. For each constituent stock i , let $\mathbf{r}_i \in \mathbb{R}^T$ be the corresponding returns and set $\mathbf{R} := [\mathbf{r}_1, \dots, \mathbf{r}_N] \in \mathbb{R}^{T \times N}$. The index tracking problem at time t consists in finding $\beta_t^* \in \mathbb{R}^N$ such that *

$$\beta_t^* := \arg \min_{\beta \in \mathbb{R}^N} \|\mathbf{r}_I - \mathbf{R}\beta\|_2^2 \quad (1)$$

$$\text{s.t. } \mathbf{0} \leq \beta \leq \mathbf{1} \quad (2)$$

$$\beta^T \mathbf{1} = 1 \quad (3)$$

$$\|\beta\|_0 = p. \quad (4)$$

Constraint (3) represents a full budget constraint, while (2) imposes a constraint on β that limits investors to taking only long positions. The cardinality constraint (4) restricts the number of stocks used to track the index to a fixed number $p \in \mathbb{N}$. An increase in p might lead to a reduction in tracking errors, possibly incurring however in larger transaction costs. Thus, p allows us to control the trade-off between tracking error and trading costs.

The cardinality constraint for β is non-convex and gives rise to an NP-hard problem. A wealth of methods exist in literature to solve this. Popular methods include applying $L_{1/2}$ regularization, using a Lagrangian version with an approximated L_0 -norm, replacing the L_0 -norm with an L_1 -norm (the LASSO technique), employing Mixed Integer Quadratic optimisation and various genetic algorithm methods. We refer to Satpathy and Shah (2022) and references therein for a more general discussion on this problem as well as an overview of these methods.

2.1.1 Optimization problem as sampling problem

In Satpathy and Shah (2022), Sequential Monte Carlo Methods (SMC) are used to solve the optimization problem 1. There, the key idea is to interpret the L_0 optimization problem as a maximization problem over the set of all possible (fixed-size) permutations of stocks composing some index and then using SMC as a global optimization technique.

Hereafter, we briefly recall the main idea in Satpathy and Shah (2022), see also Duan (2019).

The index tracking problem with a L_0 -norm penalty is a regression problem where the objective function in equation 1 is minimized. Similarly, one can consider the following maximization problem:

$$\begin{aligned}
\beta_t^* = \arg \max_{\beta \in \mathbb{R}^N} \exp(-\|\mathbf{r}_I - \mathbf{R}\beta\|_2^2) \\
\text{s.t. } \mathbf{0} \leq \beta \leq \mathbf{1} \\
\beta^T \mathbf{1} = 1 \\
\|\beta\|_0 = p.
\end{aligned} \tag{5}$$

Introducing the L_0 -norm penalty translates to finding the optimal permutation $\mathbf{P}^*$ of p stocks (a fixed number of regressors) that solves:

$$\mathbf{P}^* := \arg \max_{\mathbf{P}} \exp(-\|\mathbf{r}_I - \mathbf{R}_P \hat{\beta}_p\|_2^2), \tag{6}$$

where $\mathbf{R}_P$ is the submatrix of $\mathbf{R}$ corresponding to columns in $\mathbf{R}$ for p regressors and $\hat{\beta}_p$ is given via ¹

$$\begin{aligned}
\hat{\beta}_p := \arg \min_{\beta_p \in \mathbb{R}^p} \|\mathbf{r}_I - \mathbf{R}_P \beta_p\|_2^2 \\
\text{s.t. } \mathbf{0} \leq \beta_p \leq \mathbf{1} \\
\beta_p^T \mathbf{1} = 1.
\end{aligned} \tag{7}$$

Moreover, since for every fixed permutation $\mathbf{P}$, $\exp(-\|\mathbf{r}_I - \mathbf{R}_P \hat{\beta}_p\|_2^2) > 0$ and bounded above by 1, if a normalization constant $C > 0$ is introduced, we can interpret this as a discrete probability distribution function over the permutation space. The target distribution to be maximised is then expressed as

$$T(\mathbf{P}) := \frac{\exp(-\|\mathbf{r}_I - \mathbf{R}_P \hat{\beta}_p\|_2^2)}{C}. \tag{8}$$

The optimal composition of p stocks is then the permutation that maximizes this target distribution.

2.2 Sequential Monte Carlo Methods

A naive approach to solve problem (6) would comprise of drawing samples from $T(P)$ and choosing the optimal P that maximises it. However, $T(P)$ is not easily sampled from. For such a case, $T(P)$ can be empirically approximated via SMC methods. SMC methods sample particles ² from an initial proposal distribution

¹In Satpathy and Shah (2022) the sparse index problem is solved without considering any constraints on the $\hat{\beta}_p$. In such a case the optimal solution admits an explicit form.

²Going forward particles, samples and permutations will be used interchangeably.

after which a sequence of (sequential) importance sampling, resampling and support boosting steps are employed. A collection of weighted particles is then obtained which then represents the approximation of the target distribution. In what follows we describe in more detail the steps within SMC algorithm as employed in Satpathy and Shah (2022). See also Del Moral et al. (2006) for a more general discussion.

2.2.1 Sequential Importance Sampling

The first step in a SMC algorithm is importance sampling. It can be described as follows: N particles, denoted by $(P_i)_{i=1}^N$, are sampled from an easy-to-sample distribution $I(P)$, also called proposal distribution. Then, for each $i = 1, \dots, N$, normalized importance weights w_i are computed:

$$w_i := \frac{T(P_i)}{I(P_i)} \left(\sum_{j=1}^N \frac{T(P_j)}{I(P_j)} \right)^{-1}.$$

The pair $(P_i, w_i)_{i=1}^N$ completely characterizes the target distribution approximation and all quantities of interest from the distribution can be computed.

The quality of the sample drawn depends on the choice of $I(P)$, in particular on its closeness to the target distribution. Moreover, a necessary condition for $I(P)$ to be a good initial choice is that its support includes the support of $T(P)$.

A tool to gauge the quality of the sample (and thus of the proposal distribution) is given by the computation of its Effective Sample Size defined by:

$$ESS := \frac{1}{\sum w_i^2}.$$

Since the weights w_i are normalized, the ESS value will range from 1 to N , where we remember that N is the number of samples drawn from the proposal distribution. An ESS value of 1 indicates a poor choice for $I(P)$, contrary to a value equal to N which stands for a good choice of $I(P)$. Indeed, this happens if for instance each importance weight is equal to $\frac{1}{N}$.

However, when approximating a target distribution that is high-dimensional and complex, the simple importance sampling method tends to yield poor approximations (Satpathy and Shah (2022)). To account for this, Sequential Importance Sampling (SIS) methods are used. They aim to gradually approximate the target distribution with a sequence of approximating distributions.

SIS methods are commonly employed in filtering problems, where a sequence of new information arrives sequentially. However, in the context of this report, a sequence of new information used to approximate $T(P)$ does not occur in the same manner. To account for this, density tempering is introduced.

Density tempering is a process proposed by Del Moral et al. (2006) which allows for sampling from target distributions with higher dimensions.

Its main ideas can be summarized as follows: a collection of synthetic, intermediate target distributions $(T_{\gamma_i}(P))_{i=0}^M$ are introduced, where for all $i = 1, \dots, M$, $\gamma_i \in [0, 1]$ with $0 = \gamma_0 < \gamma_1 < \gamma_2 < \dots < \gamma_M = 1$. The sequence of γ -modulated distribution is computed as follows:

$$\forall \gamma_i, T_{\gamma_i}(P) = I(P) * \left[\frac{T(P)}{I(P)} \right]^{\gamma_i}.$$

Notice that $T_{\gamma_0} = I(P)$ and $T_{\gamma_1} = T(P)$, corresponding to the proposal and (true) target distribution. Similarly, the importance weights between each intermediate step are calculated according to the following formula:

$$w_{\gamma_j}(P) := w_{\gamma_i}(P) * \left[\frac{T(P)}{I(P)} \right]^{\gamma_j - \gamma_i}, \quad \gamma_i < \gamma_j.$$

The choice of the initial proposal distribution

The initial proposal distribution $I(P)$ suggested by Satpathy and Shah (2022) is described as follows. Consider the regression coefficient of determination R_j^2 for the j th stock returns. A high R_j^2 would imply that stock j is more likely to appear in the final P that maximises $T(P)$. A sampling strategy without replacement is considered and the probability of the first j th stock to be chosen is described as:

$$q_j = \frac{R_j^2}{\sum_{j=1}^N R_j^2}$$

Similarly, the probability for choosing the next n th stock in a set of $N - 1$ stocks is $\frac{q_n}{1 - q_k}$. Following this logic, $I(P)$ for a given permutation P of p stocks is computed as follows:

$$I(P) = \frac{q_1 * q_2 * \dots * q_p}{(1 - q_1) * (1 - (q_1 + q_2)) * \dots * (1 - \sum_{i=1}^{p-1} q_i)} \quad (9)$$

Notice that $I(P)$ described above is sequence dependent. The order in which stocks are chosen to compute $I(P)$ matters. However, this is not the case for the regression solution that we consider. Indeed, the value of $T(P)$ is the same for each combination of p stocks despite the order in which they appear within the permutation. We are only concerned with the choice of stocks used to track the index and their proportion of the current portfolio wealth (their respective weights).

2.2.2 Resampling

As $T_\gamma(P)$ evolves and with each re-weighting process more variability is introduced and fewer samples retain significant weights. This causes weight degeneracy and consequently a low ESS value. To overcome this phenomenon, one gets rid of samples with low-importance weights and repeatedly samples those with high-importance weights. This is in practice done by applying some resampling methods (see e.g. Chen (2003), Speekenbrink (2016)) whenever the ESS value falls below a certain threshold (commonly set to $N/2$). In Satpathy and Shah (2022) a multinomial resampling technique is implemented. Here, we instead consider the systematic resampling technique, which can be described as follows. Fix the number of particles to be resampled to $n \leq N$. For $k \in \{1, \dots, n\}$, let

$$u_k := \frac{(k-1) + \tilde{u}}{N} \text{ with } \tilde{u} \sim U[0, 1).$$

Replace particle k with particle i according to the following condition:

$$P_t^k = P_t^i, \text{ with } i \text{ s.t. } u_k \in \left[\sum_{j=1}^{i-1} w_t^j, \sum_{j=1}^i w_t^j \right),$$

where w^j denotes the weight corresponding to the particle j .

2.2.3 Support Boosting and Metropolis Hastings Algorithm

Since in the resampling step samples with high weights are repeatedly resampled, this method causes a loss in particle diversity, leading therefore to a shrink in the empirical support of the sequence of distributions represented by the particles and their weights. To account for this problem, an additional step is added with the intention of boosting the support. To this end, several moves of the Metropolis-Hastings (MH) algorithm are performed in Satpathy and Shah (2022). Here, we consider the same method which can be briefly described as follows: after each resampling step, MH algorithm is used for sampling from the current intermediate target distribution $T_\gamma(P)$, by using a new distribution for proposing a new sample P' and then accepting or rejecting the new sample with some probability.

The MH probability of acceptance, that is the probability of replacing the current particle P^r with the new sample P' , is computed as follows:

$$\alpha_\gamma(P^r \rightarrow P') = \min \left(1, \frac{T_\gamma(P')}{T_\gamma(P^r)} * \frac{h(P^r|P')}{h(P'|P^r)} \right),$$

where $h(P^r|P')$ denotes the proposal density.

Notice that if the ratio $\frac{T_{\gamma_i}(P')}{T_{\gamma_i}(P^r)}$ is high there is a greater chance of moving from P^r to P' .

The acceptance or rejection step in the MH algorithm is applied until the accumulative acceptance rate reaches 500% to ensure the support of the distribution is sufficiently boosted.

The choice of the proposal distribution³ in the MH algorithm

The choice of the proposal distribution h is done as follows. Initially, a count-based probability $Q_{\gamma_i}(P)$ is considered, meaning that the probability of choosing a stock to be sampled in a permutation is proportional to the number of times the stock appears in the current sample of permutations. Notice that this choice reflects the relative importance of particles after the SMC algorithm has reached the stage indicated by the current γ . Then, the proposal distribution h is defined as a weighted sum of $I(P)$, the initial proposal distribution described in Section 2.2.1, and $Q_{\gamma_i}(P)$:

$$h^\omega(P^r|P') = \omega * h_Q(P^r|P') + (1 - \omega) * h_I(P^r|P'),$$

where $\omega \in [0, 1]$. Moreover, the distribution employed in Satpathy and Shah (2022) h is defined in such a way that only a given subset A of the permutation is replaced. More precisely, to describe the computation of $h_Q(P^r|P')$ and $h_I(P^r|P')$ as it is done in Satpathy and Shah (2022), consider a subset A of the permutation P^r that we wish to replace in order to transform P^r to P' . Since only a subset of P^r is replaced, for every P' it holds that $P'_{-A} = P^r_{-A}$. Thus, we sample A from the set of stocks that exclude stocks in P^r_{-A} . In particular, if the set of all stocks is S , we sample A from $S \setminus P^r_{-A}$. We then define $h_Q(P^r|P')$ as the probability of sampling A from $S \setminus P^r_{-A}$ given the count-based probability $Q_{\gamma_i}(P)$ described above. Similarly $h_I(P^r|P')$ can be computed based on $I(P)$ applied to the set of $S \setminus P^r_{-A}$ stocks.

Summary of SMC Algorithm

Finally, we give a summary of the SMC algorithm implemented and proposed by Satpathy and Shah (2022).

1. Fix the number of stocks used to replicate the index to p .
2. Draw n permutations of size p from $T_{\gamma=0}(P) = I(P)$, for $I(P)$ computed as in Equation 9. Set the importance weights for each permutation to $\frac{1}{n}$.
3. Choose δ and increment γ such that $\gamma = \gamma + \delta$.
4. At $\gamma + \delta$, compute the importance weights for each permutation using recursive formulas in 2.2.1.

³the proposal in MH algorithm has no relation to the proposal in the Importance Sampling step.

5. Compute ESS
6. While $\gamma \leq 1$
 - (a) if $\text{ESS} \geq \frac{n}{2}$ proceed to Step 3.
 - (b) if $\text{ESS} < \frac{n}{2}$. Resample and initialise MH with an accumulative acceptance rate of 500%.
 - (c) Set the weights of new samples computed in MH to $\frac{1}{n}$ and proceed to Step 3.
7. At $\gamma = 1$, samples from $T(P)$ are obtained, represented as pairs of permutations and their respective weights.
8. Resample once more to obtain new permutations with equal weights.
9. Compute $T(P)$ for the new sample of permutations to obtain the optimal P^* that maximises $T(P)$ given by equation (8).

3 Transaction Costs

As briefly mentioned in the Introduction 1, in Satpathy and Shah (2022) transaction costs have not been accounted for when rebalancing the index tracking portfolio. This can potentially cause stocks to enter and exit the tracking portfolio with minimal benefit but at an increased cost. In order to consider this aspect, in this section we reformulate the index-tracking problem by adding an additional term to the objective function which results in a more gradual rebalancing of the index-tracking portfolio.

3.0.1 Dynamics of the wealth process given proportional transaction costs

We start by describing the evolution of the value of the index-tracking portfolio over rebalancing times. Consider the i th rebalancing time point t_i . If transaction costs are not considered, the value of the index tracking (self-financing) portfolio X , for an index comprised of N stocks, is given by:

$$X_{t_i} = X_{t_{i-1}} \sum_{j=1}^N \beta_{t_{i-1}}^j \frac{S_{t_i}^j}{S_{t_{i-1}}^j},$$

with $\beta_{t_{i-1}}^j$ representing the proportion of the wealth invested in the stock j at the previous rebalancing time t_{i-1} and only p of these weights are non-zero.

If instead transaction costs are allowed, we model the evolution of X as follows:

$$X_{t_i} = \sum_{j=1}^N \frac{\beta_{t_{i-1}}^j \tilde{X}_{t_{i-1}}}{S_{t_{i-1}}^j} S_{t_i}^j \quad (10)$$

Here, $\tilde{X}_{t_{i-1}}$ represents the value of the portfolio after the rebalancing time point t_{i-1} , $\frac{\beta_{t_{i-1}}^j \tilde{X}_{t_{i-1}}}{S_{t_{i-1}}^j}$ represents the number of shares held in stock j at time t_{i-1} . At the next rebalancing time t_i , new proportions $\beta_{t_i} = (\beta_{t_i}^1, \dots, \beta_{t_i}^N)$ of each stock comprising the current portfolio wealth is computed and the value of the portfolio needs to be rebalanced to account for this. Thus, the value of our portfolio after rebalancing is given by

$$\tilde{X}_{t_i} = X_{t_i} - C_{t_i}, \quad (11)$$

where $C_{t_i} > 0$ is the transaction cost paid for rebalancing at time t_i . Since the portfolio wealth is less due to transaction costs, we take into account that there is now less money to invest in each stock. Thus, the following equation represents the value of the portfolio after rebalancing:

$$\tilde{X}_{t_i} = \sum_{j=1}^N \beta_{t_i}^j (X_{t_i} - C_{t_i}).$$

Here, $\beta_{t_i}^j (X_{t_i} - C_{t_i})$ is the amount of money invested in the j th stock at time t_i . Recall we pay a transaction cost for each unit of money we sell or buy. Let ε be the percentage we pay per unit of money we sell or buy i.e. the rate of transaction costs. We will incur a transaction cost of ε enforced on the difference in the amount of money invested in each stock before and after the rebalancing. Following this logic the overall transaction cost should satisfy the following equation:

$$C_{t_i} = \sum_{j=1}^N \varepsilon \left| \beta_{t_i}^j (X_{t_i} - C_{t_i}) - \frac{\beta_{t_{i-1}}^j \tilde{X}_{t_{i-1}}}{S_{t_{i-1}}^j} S_{t_i}^j \right| \quad (12)$$

3.0.2 Index tracking problem with proportional transaction costs

Our goal is to minimize the transaction costs incurred at each rebalancing step, i.e. to minimize the money lost due to transaction costs, C . In order to implement a constraint on C we can reformulate the regression problem proposed by Satpathy and Shah (2022) to a regression problem with the following objective function:

$$\beta_t = \arg \min_{\beta} \|\mathbf{r}_I - \mathbf{R}\beta\|_2^2 + \lambda \|C(\beta)\|_2^2, \quad (13)$$

where $C(\beta)$ denotes the solution of equation (12) and $\lambda > 0$ is a flexible parameter needed to model the sensitivity of the algorithm to transaction costs.

Solving the modified problem (13) would require the explicit computation of the function $C(\beta)$.

Next, observe that given equation (10), (11) and (12), the evolution of X can be rewritten as

$$X_{t_i} = \sum_{j=1}^N \frac{\beta_{t_{i-1}}^j X_{t_{i-1}}}{S_{t_{i-1}}^j} S_{t_i}^j - f(S_{t_{i-1}}, S_{t_i}, \beta_{t_{i-1}}, \beta_{t_i}, \tilde{X}_{t_{i-2}}),$$

for a positive function f which represents the difference of the value of the portfolio X builds accounting for transaction costs, and the one in equation (10), where transaction costs were not considered.

Finally, for computational convenience, we assume that

$$f(S_{t_{i-1}}, S_{t_i}, \beta_{t_{i-1}}, \beta_{t_i}, \tilde{X}_{t_{i-2}}) \approx \epsilon \sum_{j=1}^N \left| \beta_{t_{i-1}}^j X_{t_{i-1}} - \beta_{t_{i-2}}^j X_{t_{i-2}} \frac{S_{t_{i-1}}^j}{S_{t_{i-2}}^j} \right|,$$

for some $0 < \epsilon \leq 1$. We then consider the following (approximated) equation describing the evolution of X :

$$X_{t_i} \approx X_{t_{i-1}} \sum_{j=1}^N \beta_{t_{i-1}}^j \frac{S_{t_i}^j}{S_{t_{i-1}}^j} - \epsilon \sum_{j=1}^N \left| \beta_{t_{i-1}}^j X_{t_{i-1}} - \beta_{t_{i-2}}^j X_{t_{i-2}} \frac{S_{t_{i-1}}^j}{S_{t_{i-2}}^j} \right|.$$

Given all these considerations, we introduce the (simplified) index tracking problem (allowing for transaction costs) as follows:

$$\begin{aligned} \beta_{t_i}^* &:= \arg \min_{\beta_{t_i}} \|\mathbf{r}_I - \mathbf{R} \beta_{t_i}\|_2^2 + \lambda \sum_{j=1}^N \left| \beta_{t_{i-1}}^j X_{t_{i-1}} - \beta_{t_{i-2}}^j X_{t_{i-2}} \frac{S_{t_{i-1}}^j}{S_{t_{i-2}}^j} \right|^2 \\ &\text{s.t. } \beta_{t_i}^T \mathbf{1} = 1 \\ &\quad \|\beta_{t_i}\|_0 = p, \end{aligned} \tag{14}$$

where $\beta_{t_{i-1}}^*$ denotes the vector of optimal weights at the previous rebalancing time. Similarly to the approach in Satpathy and Shah (2022), we introduce distribution on the space of permutations as a means to solve this problem

$$T(\mathbf{P}_{t_i}) = \exp \left(-\|\mathbf{r}_I - \mathbf{R}_{\mathbf{P}_{t_i}} \beta_{\mathbf{P}_{t_i}}\|_2^2 - \lambda \|\beta_{\mathbf{P}_{t_i}} X_{t_i} - \beta_{\mathbf{P}_{t_{i-1}}}^* X_{t_{i-1}} \frac{S_{t_i}}{S_{t_{i-1}}}\|_2^2 \right) / C,$$

where, for all $j = 1, \dots, N$ $(\beta_{\mathbf{P}_{t_{i-1}}}^* X_{t_{i-1}} \frac{S_{t_i}}{S_{t_{i-1}}})^j := \beta_{\mathbf{P}_{t_{i-1}}}^{*,j} X_{t_{i-1}} \frac{S_{t_i}^j}{S_{t_{i-1}}^j}$.

Then, applying a similar reasoning as in the case of the optimization problem without transaction costs, in order to find the optimal permutation at time t_i , $T(\mathbf{P}_{t_i})$ is approximated via SMC methods using density tempering.

4 The SMC Algorithm with Principal Component Analysis

This report expands upon the work presented in a paper by Satpathy and Shah (2022). Satpathy et al. set the number of stocks for constructing an index tracking portfolio as a constant, p representing 20% of the total number of stocks in the index. This report chooses to use Principal Component Analysis (PCA) to set the number of stocks chosen for an index tracking portfolio.

PCA is a statistical method of dimension reduction. PCA decomposes a complex data set of interrelated variables into a set of uncorrelated principal components. It achieves this by capturing the variation of the data set, while minimising the loss of information. Each principal component is derived through a linear combination of the original variables, where the relative importance of each variable is described by the coefficient terms. Principal components are ordered such that the first few components account for the most variation in the original data set. For a detailed derivation of PCA refer to Jolliffe (2002) and Yang (2015). In the context of this report, an overview of the PCA algorithm can be described as follows.

First, consider the matrix of returns over a time period for all stocks comprising an index. The return data is centered and the covariance matrix is then calculated. Next, PCA computes the corresponding eigenvalues and eigenvectors of the covariance matrix. The reduced dimension matrix of principal components, $\mathbf{Z}$ is then computed using the following equation:

$$\mathbf{Z} = \mathbf{A}^T \mathbf{R}$$

Matrix $\mathbf{A}$ is defined as orthogonal, and its k^{th} column corresponds to the k^{th} eigenvector of the covariance matrix. The eigenvectors are sorted in descending order, with the first eigenvector representing the one associated with the highest eigenvalue. The number of eigenvectors used to construct $\mathbf{A}$ is determined by the desired level of variance explained by the principal components, effectively acting as a constraint on the dimension of the reduced matrix.

PCA is implemented using the Python package `scikit-learn` and the variance explained by the principal components is set to 90%. The variable p is then assigned a value equal to the number of principal components computed under this constraint.

5 Methodology and Data

This section contains the details of the numerical experiments performed in this study.

5.1 Models

In this study, the index-tracking algorithm utilized builds upon the algorithm outlined in Section 2.2.3, incorporating an extension that involves employing Principal Component Analysis (PCA) to determine the number of stocks in the tracking portfolio. The models examined in this analysis primarily vary in their hyperparameter settings.

To provide a concise overview of the diverse models formulated, the table presented below summarizes the hyperparameters employed for each model.

Hyper-parameters	Values
Step Size (δ)	0.5
Lookback Window Size (T)	30
ESS Threshold	$\frac{N}{2}$
Number of Particles (n)	100
Transition Kernel Weight (ω)	$\omega \sim U(0.2, 0.8)$
Cumulative Acceptance Rate	500%
Variance Explained	95%
Rebalance Frequency (Observations)	60
Transaction Cost per Unit Nominal (ϵ)	0.05
Transaction Cost Penalty (λ)	$\{0, 1, 10\}$

Table 1: Table of Hyper-parameters Chosen for Implementation and Comparison of Models on Real World Data.

In order to assess the performance of the updated model that incorporates transaction costs compared to the initial model, we maintain the hyperparameters of both models at the values specified in Table 1. When $\lambda = 0$, this corresponds to the initial model, without accounting for transaction costs.

It is important to note that we randomly select weights (ω) within the range of 0.2 to 0.8 for the transition kernel. The introduction of random weights (ω) within the range of 0.2 to 0.8 in the transition kernel serves a specific purpose in the algorithm. This deliberate choice aims to inject a certain level of noise into the Metropolis-Hastings framework, with the intention of enhancing the support boosting step of the algorithm. By incorporating this randomization, we aim to explore a wider range of potential solutions, potentially uncovering more favorable outcomes and

improving the overall performance of the tracking model. This approach acknowledges the potential benefits of introducing variability and perturbation into the algorithm, providing an opportunity for further optimization and robustness in the pursuit of accurate index tracking.

5.2 Constraints and Optimisation

In our numerical experiments, we implemented the index-tracking algorithm with the inclusion of no-short selling constraints and full-budget constraints. It is worth noting the implications of introducing these constraints. When incorporating short-selling and full-budget constraints, the explicit solution of the β parameters becomes infeasible, necessitating the use of numerical optimization techniques to address this challenge. This introduces a critical consideration, as the numerical optimization algorithm needs to be executed multiple times during the execution of the index-tracking algorithm, particularly when the Effective Sample Size (ESS) falls below the threshold, requiring Metropolis-Hastings steps. Consequently, this can significantly increase the overall runtime and potentially limit the algorithm's performance and efficiency.

Fortunately, the problem at hand benefits from the presence of a closed-form expression for the Jacobian, which provides valuable gradient information. The closed-form expression for the Jacobian, obtained by simply differentiating equation (14), is given by

$$(R^T R)\beta - R^T r_I + 2\lambda(X_{t_i} - \beta_{t_{i-1}}^* X_{t_{i-1}} \frac{S_{t_i}}{S_{t_{i-1}}}).$$

Numerical optimizers that utilize this gradient information can leverage it to enhance the convergence speed and improve overall efficiency. In our study, we employed the Sequential Least Squares Quadratic Programming (SLSQP) algorithm while incorporating the provided Jacobian. This strategic approach allowed us to reduce the computation time by a substantial factor of approximately 7.

The utilization of the SLSQP algorithm in conjunction with the available Jacobian information proved to be a valuable optimization strategy for our algorithm. By leveraging the closed-form expression of the Jacobian, the algorithm benefited from accelerated convergence, resulting in a significant reduction in computation time. This optimization enhancement is particularly advantageous given the necessity of executing the numerical optimization algorithm multiple times throughout the index-tracking process, thereby enabling more efficient and timely execution.

The successful reduction in computation time achieved through the incorporation of the SLSQP algorithm and the utilization of the Jacobian underscores the importance of employing suitable optimization techniques in addressing the computa-

tional challenges associated with complex constraints. This optimization strategy contributes to enhancing the algorithm's performance, enabling it to handle larger-scale index-tracking problems efficiently and effectively.

5.3 Data

Simulated Data

In the process of algorithm development, we employed simulated index and stock data where the true β parameters were known. To generate the stock price paths for N stocks up to some future time $T > 0$, we employed the Standard Geometric Brownian Motion (GBM) model. Each stock price path follows the process

$$dS_t^{(i)} = \mu_i S_t^{(i)} dt + \sigma_i S_t^{(i)} dW_t^{(i)}$$

where μ_i and σ_i denotes the drift and volatility of the i -th stock, respectively, and $\{W_t^{(i)} : t \geq 0\}$ denotes a standard Brownian Motion process. The Brownian Motion processes are correlated such that $\mathbb{E}(dW_t^{(i)} dW_t^{(j)}) = \rho_{i,j} dt$ and $\rho_{i,i} = 1$. This leads to an explicit form to determine the stock price paths for each stock $i = 1, \dots, N$ using

$$S_{t_k}^{(i)} = S_0^{(i)} \exp \left(\sum_{j=1}^k \left(\mu_i - \frac{1}{2} \sigma_i^2 \right) \Delta t_j + \sqrt{\Delta t_j} \sigma_i X_i \right)$$

with time points $0 = t_0, t_1, \dots, t_n = T$ and time increments $\Delta t_j = t_j - t_{j-1}$. X_i corresponds to the i -th element in a multivariate normal vector $\mathbf{X} \sim \mathcal{N}_N(\mathbf{0}, \boldsymbol{\rho})$ where $\boldsymbol{\rho} = \text{Corr}(\mathbf{X}, \mathbf{X}')$ denotes the correlation matrix. The multivariate normal random vectors can be calculated as $\mathbf{X} = \mathbf{L}\mathbf{Z}$ where $\mathbf{Z} \sim \mathcal{N}_N(\mathbf{0}, \mathbf{I})$ and $\mathbf{L}$ denotes the Cholesky decomposition of the correlation matrix $\boldsymbol{\rho}$.

To account for the dynamic nature of index constituents, we established a fixed maximum number of stocks (N) in the asset universe and a maximum number of stocks that could compose the index ($N_I \leq N$). From the asset universe $\Omega = \{1, \dots, N\}$, a random selection of stocks was made to form the index, along with random time points indicating when these stocks entered or exited the index. This gives the set of stocks that the index comprises of at each time point as $I_t \subseteq \Omega$. This simulation setup mirrors the dynamic nature of real-world indices, where stocks are added or removed based on specific criteria and timing.

In order to construct the simulated index time series, we take the weighted average of stock returns in the index at each time point, i.e.,

$$r_{I_t} = \sum_{i \in I_t} \beta_t^{(i)} R_t^{(i)}$$

where at each time point we have that

$$\sum_{i \in I_t} \beta_t^{(i)} = 1$$

The simulated data spanned a predetermined temporal window, with a daily frequency. The parameters for drift, volatility, and initial stock prices for the constituent stocks in the index were uniformly sampled within appropriate ranges. The rationale behind uniformly sampling GBM parameters was primarily to introduce random variation among the simulated stocks. By incorporating this variation, we aimed to capture the inherent diversity and unpredictability observed in real-world stock markets. This approach ensured that the simulated data reflected a realistic scenario, allowing us to evaluate the algorithm’s performance under more representative conditions.

To evaluate the accuracy and effectiveness of the index-tracking algorithm, we calculated index returns as the average of the individual stock returns that constituted it (i.e. $\beta_t^{(i)} = \frac{1}{|I_t|}$). By aggregating these returns, we obtained the index value, which served as a benchmark for evaluating the algorithm’s ability to accurately select the constituent stocks and estimate the β parameters.

Throughout the simulation experiments, the index-tracking algorithm consistently demonstrated the capability to effectively identify the stocks forming the index and accurately estimate the associated β parameters. This successful performance suggests that the algorithm holds promise in real-world scenarios, where the selection of appropriate stocks and the estimation of β parameters are crucial for achieving accurate index tracking.

The primary objective of the simulation study was twofold: first, to verify the efficacy of the index-tracking algorithm and second, to refine and build upon the algorithm itself. As such, the focus of this study was primarily on algorithm development and validation rather than presenting and analyzing specific simulation results. By prioritizing the algorithm’s performance under controlled simulated conditions, we aimed to establish its robustness and reliability.

Therefore, in order to emphasize the algorithm’s performance in real-world scenarios and its applicability to actual data, the decision was made to not present detailed simulation results in this context. Instead, the focus shifts to the application of the algorithm to real-world data, where its effectiveness can be better evaluated and its potential impact on practical index tracking can be assessed. The transition from simulated data to real-world data allows for a more meaningful and relevant analysis, providing valuable insights into the algorithm’s practical performance and potential benefits.

Empirical Data

For the empirical study, we have extracted various indices from Bloomberg for different geographical locations. This resulted in a diverse set of empirical data sets with different observation lengths and number of constituent stocks. The results presented in this paper is based on the following indices:

Index	Number of Observations	Number of Stocks
UKX	253	101
HSI	579	81
JALSH	373	131

Table 2: Table of Equity Indices used to test the Performance of the Algorithm.

We consider a combination of indices, both with a large and a small number of constituent stocks. The daily prices for the constituent stocks, as well as the corresponding values of that index are collected from a Bloomberg terminal. Dates for stock and index returns are aligned with each other. We only consider time periods in which the data exists for both stocks and indices to ensure the data is complete. Historically some indices were originally constituted by a smaller number of stocks. Moreover, over time certain stocks enter or exit an index. Hence, for numerical and programming reasons, we decided to use time intervals when full information about all the stocks included in the index is available.

5.4 Performance Metrics

Tracking error

In our study, the primary performance measure we utilize is the tracking error, as defined in Satpathy and Shah (2022). For actual index returns r_{I_t} and tracking portfolio returns $\beta'_t \mathbf{R}_t$ at times $t = 1, \dots, T$, the tracking error can be calculated as

$$\text{TE} = \sqrt{\frac{1}{T-1} \sum_{t=1}^T (r_{I_t} - \beta'_t \mathbf{R}_t)^2} \quad (15)$$

The tracking error quantifies the level of deviation between the returns of the index tracking portfolio and the target index. It can be calculated as the standard deviation of the return residuals, which are the differences between the returns of the index and the returns of the index tracking portfolio.

By examining the tracking error, we gain insights into the accuracy and effectiveness of the index tracking strategy. A lower tracking error indicates a closer alignment between the portfolio's performance and the target index, suggesting a higher degree of replication. Conversely, a higher tracking error implies a greater level of deviation, indicating potential inefficiencies in the tracking strategy.

Total Transaction Costs

In the context of transaction cost-conscious investing, monitoring the evolution of the wealth process over time becomes crucial. Therefore, as a second equally important metric, we focus on the total transaction costs incurred throughout the investment time horizon. If transaction costs are defined as in section 3, then the total transaction costs can be calculated as

$$TC = \sum_{i=1}^n C_{t_i} \quad (16)$$

Tracking the total transaction costs allows us to assess the overall cost-effectiveness of the tracking strategy. These costs are calculated at the end of each rebalancing period, representing the expenses incurred during the adjustment of the portfolio's composition. By analyzing the total transaction costs at the end of the investment period, we gain valuable insights into the performance of the tracking strategy in terms of transaction costs.

The interpretation of the results regarding transaction costs should take into account the specific investment objectives and constraints of the investor. Lower total transaction costs are generally desired, as they indicate a more efficient allocation of resources. However, it is important to balance these cost considerations with the tracking error. Intuitively, there is often a trade-off between minimizing tracking error and minimizing transaction costs. Achieving a lower tracking error may require more frequent portfolio rebalancing, leading to higher transaction costs. Conversely, reducing transaction costs may involve a less frequent rebalancing strategy, potentially resulting in a higher tracking error.

By evaluating both the tracking error and total transaction costs, we can comprehensively assess the performance of the index tracking strategy. It allows us to understand the trade-offs between achieving accurate index replication and managing transaction costs, enabling investors to make informed decisions based on their specific investment goals and preferences.

6 Results

We first start with showing numerical results of our experiments and then focus on illustrations:

Cost Penalty	TC	TE	Stocks	Observations	Index
0	0.25605034	0.0111446	101	253	UKX
1	0.25456173	0.01083406	101	253	UKX
10	0.21098796	0.00973762	101	253	UKX
0	0.34191537	0.01302433	131	373	JALSH
1	0.32203768	0.0123977	131	373	JALSH
10	0.30493293	0.0116939	131	373	JALSH
0	0.47397364	0.01310602	81	579	HSI
1	0.45218308	0.01332093	81	579	HSI
10	0.416406	0.01171881	81	579	HSI

Table 3: Empirical Performance Metrics

Here the "TC" column denotes the overall transaction costs (see equation (16)) and "TE" denotes the tracking error which was introduced previously in equation (15). Table 3 demonstrates the overall performance of different models across 3 market indices. It can be seen that the higher penalty terms on transaction costs tend to decrease overall transaction costs for the majority of indices, aligning with our expectations. This observation is noteworthy as it indicates that the imposition of penalties on transaction costs successfully achieves the intended objective of reducing overall expenses.

Surprisingly, the degree of penalty imposed on transaction costs does not significantly influence the tracking error. In all cases, the tracking strategies characterized by the highest cost penalties outperform other models in terms of both costs and tracking error. This intriguing finding suggests that the relationship between transaction costs and tracking error is more intricate and multifaceted than initially presumed.

Overall, the results suggest that higher penalty terms on transaction costs can effectively reduce overall expenses, while their impact on tracking error remains relatively independent of the imposed penalties. These insights shed light on the complex interplay between transaction costs and tracking error, highlighting the need for further investigation into the underlying mechanisms that drive their relationship.

Now we will look at the plots demonstrating the performance of the prepared models:

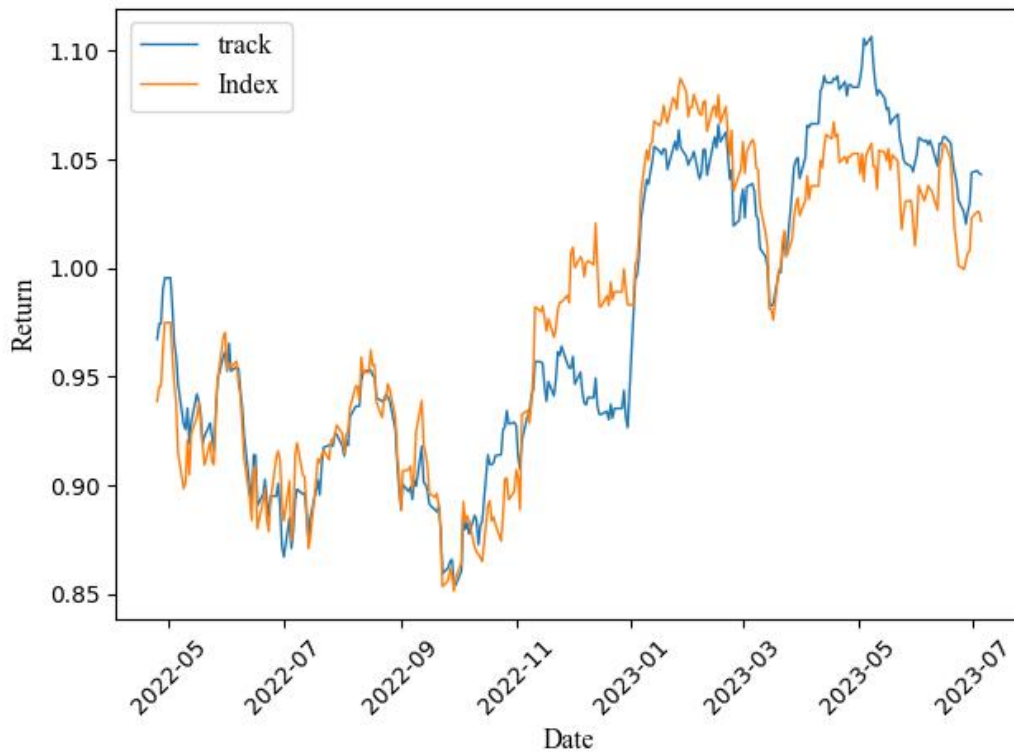


Figure 1: JALSH Tracking portfolio value versus actual index price.

By looking at 1 we can see how the learned strategy copes with tracking the real index. This part replicates the work of Satpathy and Shah (2022). Other similar plots can be found in the Appendix 7



Figure 2: UKX Tracking Portfolio Wealth Process for Different Cost Penalties

Figure 2 demonstrates how the tracking portfolio evolves under different strategies. As we can see, the strategy based on the model with the highest transaction costs penalty outperforms other strategies, that is, has the smallest cumulative expenses.

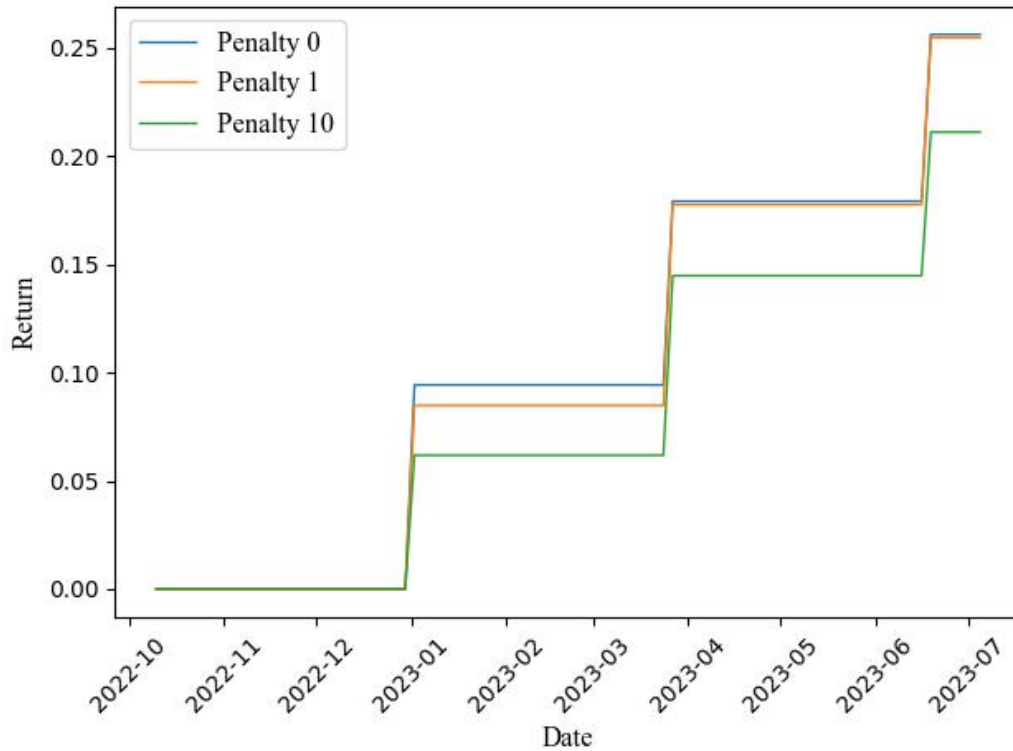


Figure 3: UKX Cumulative Transaction Costs for Different Cost Penalties

The plot above 3 illustrates the effect of varying the cost penalty. Clearly, the transaction costs are less for higher cost penalties, as expected.

The analysis of higher penalty terms on transaction costs yields several noteworthy conclusions. Firstly, it is observed that implementing such penalties successfully reduces the overall transaction costs for most indices, aligning with the expected outcome. This indicates that the imposition of penalties effectively achieves the objective of mitigating expenses.

Surprisingly, the degree of penalty imposed on transaction costs has minimal impact on the tracking error. Regardless of the penalties imposed, the tracking strategies with the highest cost penalties consistently outperform other hyperparameterizations in terms of both costs and tracking error. This intriguing finding suggests that the relationship between transaction costs and tracking error is more intricate and nuanced than initially assumed.

These conclusions emphasize the complex interplay between tracking error minimization and the incorporation of transaction costs into the objective function. The study reveals that there are additional factors at play beyond the straightforward

trade-off between minimizing tracking error and integrating transaction costs. This highlights the need for a deeper understanding of the underlying mechanisms that govern the relationship between these variables.

Importantly, the findings demonstrate that it is possible to consider transaction costs while still achieving effective tracking of an index. This challenges the notion that incorporating transaction cost considerations compromises the ability to closely track a benchmark. Instead, the results suggest that investors and portfolio managers can integrate transaction cost considerations into their strategies without sacrificing the goal of closely tracking an index.

Overall, these conclusions underscore the significance of further research and analysis to fully grasp the intricate dynamics between transaction costs, tracking error, and portfolio optimization. The report provides insights that challenge initial assumptions and highlight the potential for refining strategies that simultaneously account for transaction costs and maintain effective index tracking.

An additional noteworthy observation is the utilization of the Metropolis-Hastings algorithm in the proposed method, which introduces inherent stochasticity into the runtime. This stochastic nature stems from the Monte Carlo method employed within the Metropolis-Hastings algorithm.

Monte Carlo methods rely on random sampling to approximate solutions, resulting in a certain level of variability in the runtime of the algorithm. Even when provided with the same input data, the number of iterations necessary for convergence, the acceptance rate of proposed moves, and other factors contribute to the stochastic runtime.

The stochastic nature of the algorithm's runtime implies that each execution may yield slightly different results, despite the consistent input data. This randomness should be considered when interpreting the outcomes of the algorithm. It emphasizes the importance of conducting multiple runs or employing appropriate statistical techniques to account for the inherent variability and ensure robust conclusions.

By acknowledging the stochasticity introduced by the Monte Carlo method and the Metropolis-Hastings algorithm, we can better understand the potential variability in the runtime and results of the proposed method. This awareness prompts the need for careful analysis and proper statistical considerations when interpreting and drawing conclusions from the algorithm's outcomes.

7 Conclusions

The analysis of higher penalty terms on transaction costs yields several noteworthy conclusions. Firstly, it is observed that implementing such penalties successfully reduces the overall transaction costs for most indices, aligning with the expected outcome. This indicates that the imposition of penalties effectively achieves the objective of mitigating expenses.

Surprisingly, the degree of penalty imposed on transaction costs has minimal impact on the tracking error. Regardless of the penalties imposed, the tracking strategies with the highest cost penalties consistently outperform other hyperparameterizations in terms of both costs and tracking error. This intriguing finding suggests that the relationship between transaction costs and tracking error is more intricate and nuanced than initially assumed.

These conclusions emphasize the complex interplay between tracking error minimization and the incorporation of transaction costs into the objective function. The study reveals that there are additional factors at play beyond the straightforward trade-off between minimizing tracking error and integrating transaction costs. This highlights the need for a deeper understanding of the underlying mechanisms that govern the relationship between these variables.

Importantly, the findings demonstrate that it is possible to consider transaction costs while still achieving effective tracking of an index. This challenges the notion that incorporating transaction cost considerations compromises the ability to closely track a benchmark. Instead, the results suggest that investors and portfolio managers can integrate transaction cost considerations into their strategies without sacrificing the goal of closely tracking an index.

Opportunities for Further Study

Future research to be explored could include:

1. Consider the market impact as it can also lead to the additional losses when working with large volumes. Hence, investor should come up with smart order routing algorithms to optimize trade execution and reduce market impact. These algorithms can help split large orders into smaller ones and execute them in a way that minimizes price impact.
2. Consider liquidity screening to avoid investing in assets that may have high transaction costs or are illiquid, as these can significantly impact the performance of your tracking strategy.
3. Consider short sales so one can also track "short" indices. As well as considering short position, i.e. negative β 's which will relax the constraints in the

proposed method.

4. Consider other types of transaction costs: Tiered Commissions, Fixed-Plus-Percentage Commissions, Flat Fee Commissions.

Bibliography

- Barber, B., Odean, T., 2000. Trading is hazardous to your wealth: The common stock investment performance of individual investors. *Journal of Finance* 55, 773–806.
- Benidis, K., Feng, Y., Palomar, D.P., et al., 2018. Optimization methods for financial index tracking: From theory to practice. *Foundations and Trends® in Optimization* 3, 171–279.
- Chen, Z., 2003. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics* 182, 1–69.
- Del Moral, P., Doucet, A., Jasra, A., 2006. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 68, 411–436.
- Duan, J., 2019. Variable selection with big data based on zero norm and via sequential monte carlo. *CompSciRN: Computational (Topic)* .
- Jolliffe, I.T., 2002. Principal component analysis for special types of data. Springer.
- Rompotis, G.G., 2013. Actively vs. passively managed exchange traded funds. *Aestimatio* , 116–135.
- Satpathy, T., Shah, R., 2022. Sparse index tracking using sequential Monte Carlo. *Quantitative Finance* 22, 1579–1592.
- Speekenbrink, M., 2016. A tutorial on particle filters. *Journal of Mathematical Psychology* 73, 140–152.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58, 267–288.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention, in: *International conference on machine learning*, PMLR. pp. 2048–2057.
- Yang, L., 2015. An Application of Principal Component Analysis to Stock Portfolio Management .

Appendix

In this part we will provide some additional plots similar to those given in 5.

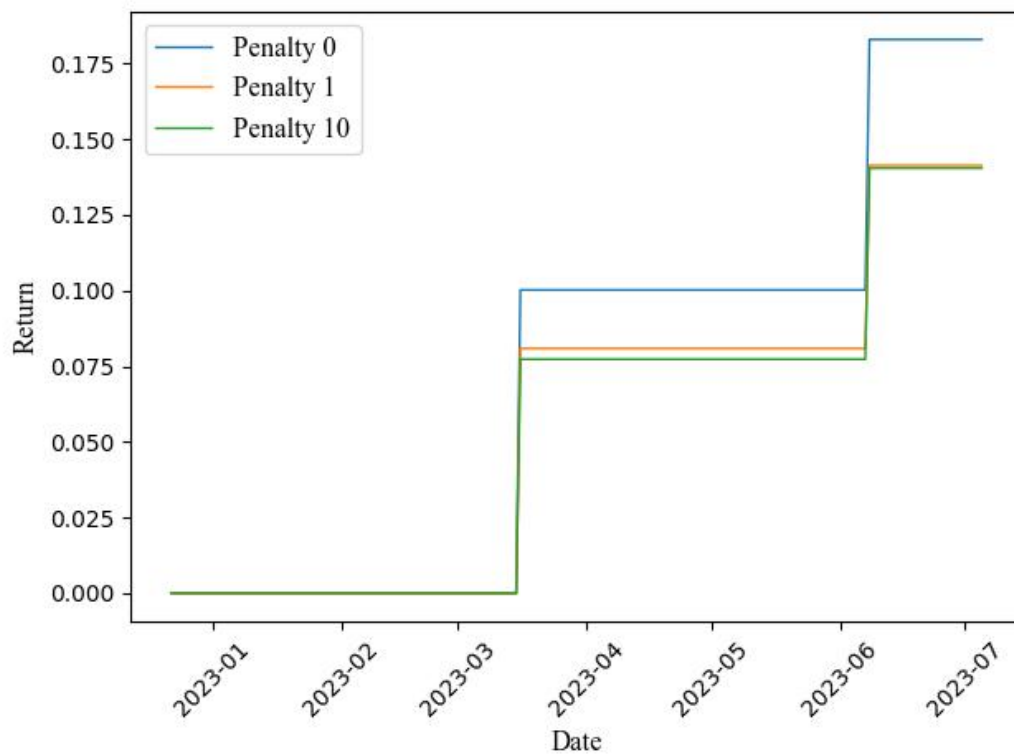


Figure 4: DAX Cumulative Transaction Costs for Different Cost Penalties

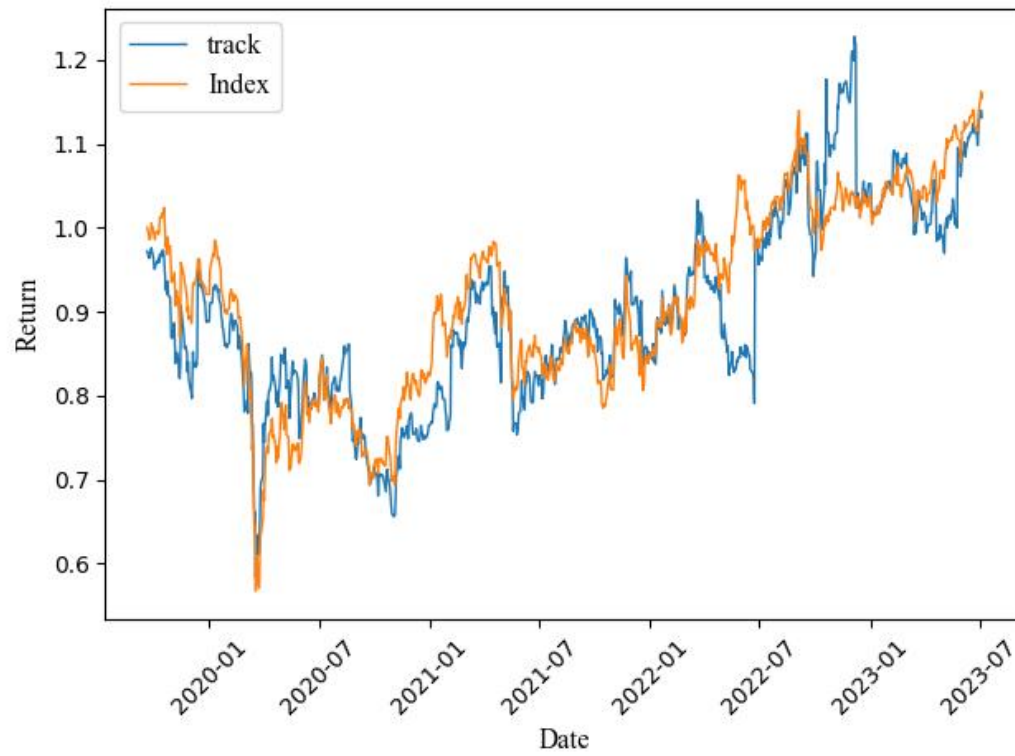


Figure 5: IPSA Tracking portfolio value versus actual index price

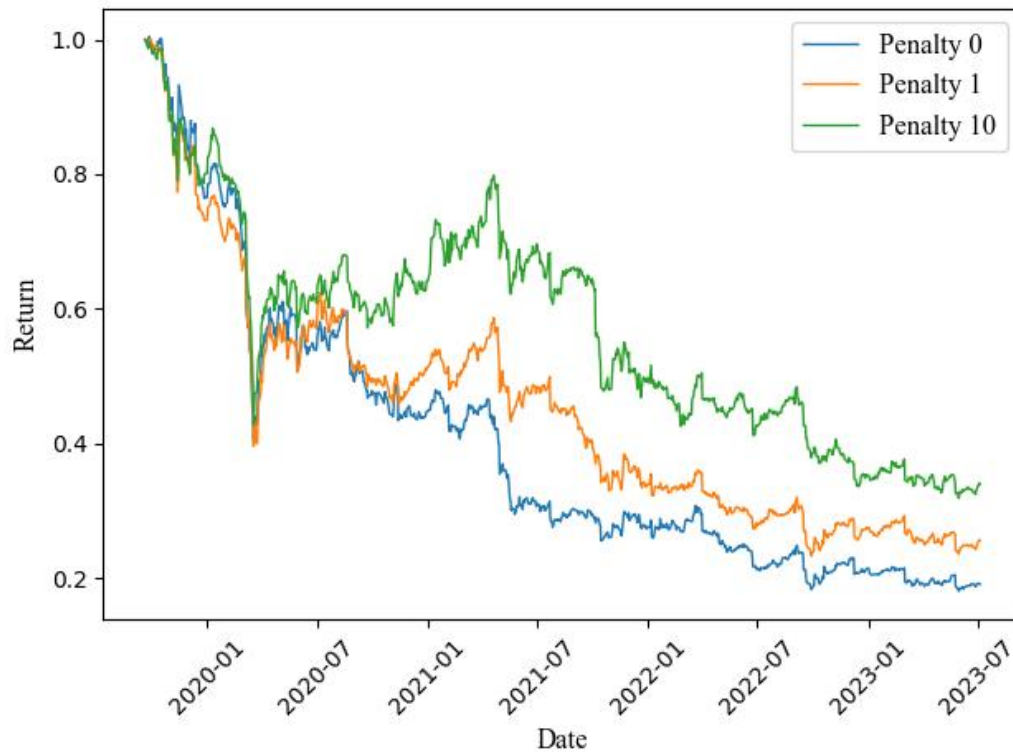


Figure 6: IPSA Tracking Portfolio Wealth Process for Different Cost Penalties



Figure 7: IPSA Tracking portfolio value versus actual index price

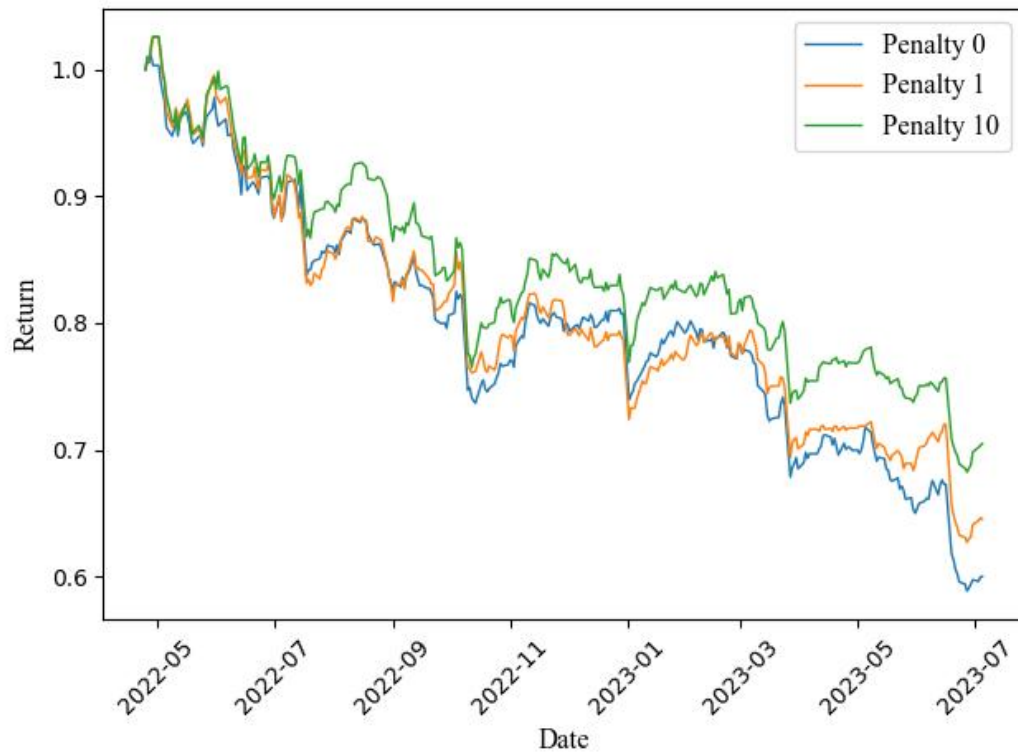


Figure 8: JALSH Tracking Portfolio Wealth Process for Different Cost Penalties

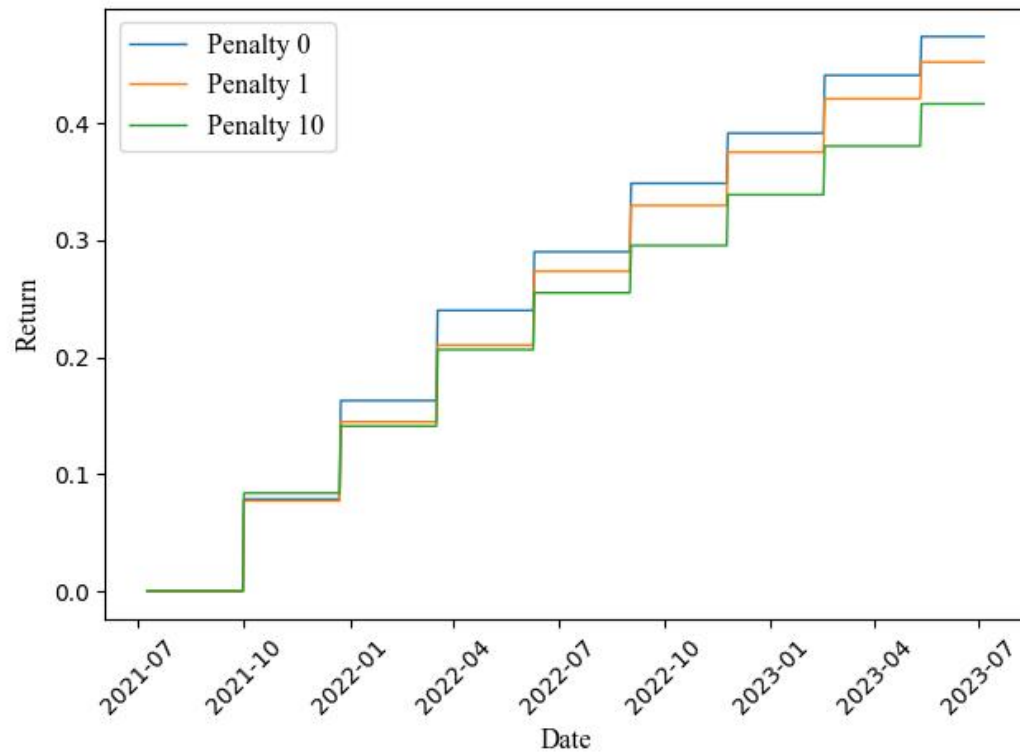


Figure 9: HSI Cumulative Transaction Costs for Different Cost Penalties

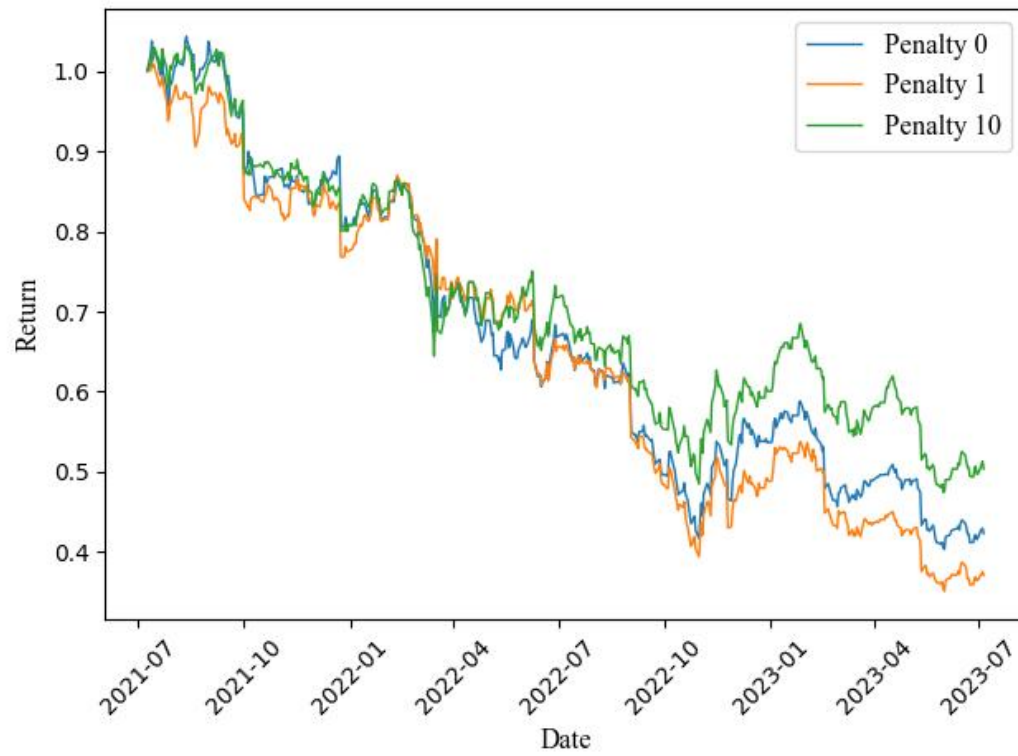


Figure 10: HSI Tracking Portfolio Wealth Process for Different Cost Penalties



Figure 11: HSI Tracking portfolio value versus actual index price

South Africa's Carbon Opportunity

TEAM 4

EBA HAMPWAYE, University of Cape Town
SIPHOKAZI HLALUKANA, University of Cape Town
FLORIAN KRACH, ETH Zurich
NCHAKHA THATO RATEELE, University of Cape Town

Supervisors:

ASHLEY KANTER, Riskworx
ANDREA MACRINA, UCL & University of Cape Town

Contents

1	Introduction	5
2	Background	7
2.1	Environmental, social and governance (ESG) framework . . .	7
2.2	The South African Context	8
2.3	Climate Change Risks (Transmission Channels)	10
2.4	Network for Greening the Financial System (NGFS)-Scenarios	11
3	Model Assumptions	12
3.1	Assumptions of the Classical Merton	12
3.2	Assets and Liabilities Model	13
3.3	Minimum Electricity Price to Cover the Total Costs	14
3.4	Probability of Default and Running Probability of Default . .	14
4	Model Example	15
4.1	Eskom Overview	15
4.2	Data Description	17
5	Results and Discussion	21
5.1	Parameters and Considered Scenarios	21
5.1.1	Base Case: IRP 2019	22
5.1.2	Second Case: IRP-2019 with "Green Continuation" .	24
5.1.3	Third Case: IRP-2019 with "Aggressive Green Con- tinuation"	25
5.1.4	Fourth Case: IRP-2019 with "Green Continuation with higher Electricity Prices"	25
5.2	Discussion	25
5.2.1	Capacities Energy Mix	25
5.2.2	Running Default Probabilities for Different Cases. .	26
5.2.3	Minimum Electricity Prices to Cover Costs	27
6	Valuation adjustment	29
6.1	Pricing interest rate swaps	30
6.2	Credit Valuation Adjustment	31
6.3	Carbon valuation adjustment	32
6.3.1	Replication pricing	33
6.4	Eskom IRS Example	37

7	Conclusions	39
---	-----------------------	----

Disclaimer

This report is a personal view and does not represent the views of any organisation such as a private company or academic institution. This report is no advice or guidance. Certain information contained in this document has been obtained or derived from third party sources and such information is believed to be correct and reliable but has not been independently verified. Furthermore the information may not be current due to, among other things, changes in the financial markets or economic environment. No obligation is accepted to update any such information contained in this presentation. The authors and their affiliations shall not be liable in any manner whatsoever for any consequences or loss (including but not limited to any direct, indirect or consequential loss, loss of profits and damages) arising from any reliance on or usage of any material contained in this report and accepts no legal or other responsibility to any party who directly or indirectly views or receives this material.

1 Introduction

Climate change results from a shift in climate patterns. This shift has been accelerated by human action through the release of greenhouse gases (GHGs) into the atmosphere (Hayhoe et al., 2018). Globally, the electricity generation sector is responsible for 43% of carbon emissions consisting of carbon dioxide (CO₂) and methane (CH₄) in 2020, see Climate Watch (2020).

Many countries, financial institutions, and organisations are in support of facilitating the reduction of carbon emissions. Nearly 200 countries signed the Paris Agreement Act of 2015 to slow down climate change by limiting the rise in temperature to 2°C above the pre-industrial levels, and ideally, 1.5°C (Blaufelder et al., 2021). It was further suggested that this requires that GHG levels are cut by 50% in 2030 and net zero be achieved by 2050. As a result, countries and companies have implemented various strategies to reduce emissions. These approaches encompass taxation, provision of subsidies, and the voluntary carbon credits market, all of which have significant implications for financial firms and their products.

In response to the climate change issue, nations are increasingly turning to the implementation of carbon taxes. A carbon tax puts a price on carbon by stating the taxation rate on GHG emissions above an established standard or on the amount of carbon present in fossil fuels that exceeds the pivot point (The World Bank, 2023). Therefore, the price of carbon is driven by a central government. The tax collected from suppliers will lead to an increase in prices for finished goods or services, which consumers will ultimately bear the cost of. Such incentives/penalisation impacts both, suppliers and consumers to move towards low-carbon/renewable sources. This point is supported by Parry (2019) who suggests that a \$35 per ton carbon tax would reduce emissions by a larger percentage than the Paris pledge in the Group of Twenty (G20) countries, shown in Figure 1.

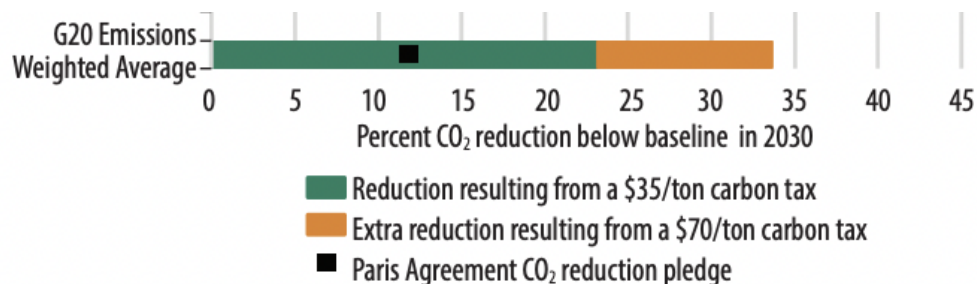


Figure 1: G20 GHG Emission Reduction with Introduction of Carbon Tax (Parry, 2019).

There is a rising concern that certain companies might relocate their production to countries where carbon taxes are not mandated, with the intention of importing the finished goods and thereby avoiding the need to pay carbon taxes. This is known as carbon leakage. Many countries, including the EU, are proposing a carbon border adjustment, which places charges on the carbon content of imported products to ensure treatment equivalent to domestic carbon pricing, potentially combined with rebates for the carbon content of exports (Keen et al., 2022). The charge can be interpreted either as a tax or as a requirement for companies importing goods to purchase allowance for carbon emissions during the production of the good (European Commission, 2023). The benefit of this adjustment is that it mitigates the risk of carbon leakage and increases the likelihood of implementing carbon taxes in export countries that currently do not have them, as the cost of importing their goods would be lower.

Other subsidies include incentives given by the government such as grants and tax rebates (Gandhi and Cuervo, 1998) directed at companies or individuals that use clean technologies in production or produce negative emissions (absorb CO₂). This approach ensures that the burden of GHG emissions falls on heavy polluters, as the revenue generated from carbon taxes can be utilized to fund such subsidies.

Lastly, carbon credits, which are purchased voluntarily from the carbon market, also work to support GHG emission reduction. Blaufelder et al. (2021) define carbon credits as certificates constituting the quantities of GHGs that have been kept out or removed from the atmosphere. To meet the world's targets, Blaufelder et al. (2021) estimated a demand increase in carbon credits of up to 7 to 13 gigatons (Gt) of CO₂ by 2050 under the Network for Greening the Financial System (NGFS) 1.5°C and 2°C scenarios. The demand and supply of the certificates influence the prices per ton. In the certificates market, the government issues new certificates, whereas in the carbon futures market, previously issued certificates are traded among companies. A carbon credit future is an instrument that is entered into by two parties, where the underlying is a carbon credit, allowing the buyer to offset their emissions without directly investing in negative-emission projects (DGB Group, 2023). This means that entering this contract reduces the risk of fluctuating prices of carbon credits as they are dependent on demand and supply. Carbon futures markets, e.g., the EUA, are young and limited with expiry dates up to a couple of years.

The above are financial initiatives that have been undertaken to address some of the man-made impacts driving climate change. However, the incorporation of carbon emissions into project financing faces a major challenge due to the limited availability of information. This report attempts to address this challenge by bridging the existing information gaps and offering valuable insights. The objective is to facilitate the integration of carbon emissions into the pricing and risk management

of financial instruments related to a power utility firm. By subjecting various simulations to stress testing against the scenarios defined by the NGFS, explained in Section 2.4, we assess the probability of default and market prices of bonds susceptible to emissions-related risks. It is important to note that NGFS scenarios are not derived from historical data nor do they serve as forecasts/predictions of future outcomes. Instead, they represent potential future outcomes, allowing us to evaluate and prepare for a range of possibilities.

The report is structured as follows. Risks associated with climate change are briefly discussed in Chapter 2, along with the measures taken to reduce these risks. The theoretical foundation of structural models as well as the assumptions made for the model we propose are explained in Chapter 3. In this report we consider Eskom, South Africa's state-owned power utility firm. In Chapter 4 we give a general review of Eskom, a description of the data we used, and instructions on how to test the suggested models. In Chapter 5 we present our results when applying our proposed modelling framework to Eskom and discuss the effects of the different NGFS scenarios on Eskom's viability. Moreover, in Chapter 6, we discuss the value adjustment for carbon costs of interest rate swaps linked to Eskom.

2 Background

2.1 Environmental, social and governance (ESG) framework

Commercial investment strategies usually revolve around shareholder capitalism, prioritising the maximisation of profits and share prices. ESG investing goes beyond this by incorporating other factors that include environmental, social, and governance challenges and developments in the analysis, selections and management of investments. (Boffo and Patalano, 2020). Inderst and Stewart (2018) expand these factors as follows:

- **Environmental:** climate change, carbon emissions, pollution, resource efficiency, biodiversity;
- **Social:** human rights, health and safety, diversity policies, community relations;
- **Governance:** corporate governance, corruption, rule of law, institutional strength, and transparency.

Climate change is a significant factor that poses challenges to achieving sustainability. The importance of corporate sustainability has been steadily increasing and is now a major concern for many businesses. Oprean-Stan et al. (2020) delves into this idea, highlighting that while the maximisation of profits and share prices remains crucial, the factors associated with sustainability, such as addressing climate

change, have an impact on long-term sustainable development. This perspective emphasises that solely focusing on short-term factors fails to account for the future implications and necessitates the inclusion of long-term factors through the ESG framework.

However, the challenge associated with incorporating the ESG framework is the lack of a universally accepted taxonomy. The distinction between what qualifies as socially and environmentally sustainable can vary across different communities, countries, and regions. This variability poses difficulties in establishing a definitive measure of what is considered sustainable, clean, or green. To address the challenge, this report employs the concept of emission footprint, which is measured in the carbon dioxide equivalence, CO₂e. The emission footprint refers to the amount of GHGs emitted by a company or project activities, converted to a unifying metric, given in terms of CO₂e. This approach serves the purpose of standardising the measurement process, enabling improved accuracy and easier comparability across different entities during analysis. In particular, this approach follows a sequence of recent papers (Kenyon et al., 2021, 2022, 2023), where the authors proposed the so-called carbon equivalence principle for the re-design and analysis of all financial products so that carbon flows are treated in the same way as cash flows in financial terms sheets.

2.2 The South African Context

South Africa is one of the countries that signed the Paris Agreement Act of 2015, showing its support in the reduction of GHG emissions according to South Africa's Low Emissions Development Strategy (SA-LEDS). The updated 2030 Nationally Determined Contribution (NDC) target for South Africa is 366–436 MtCO₂e while net zero-emissions is the 2050 NDC target (Climate Action Tracker, 2023). Figure 2 shows the current standing of South Africa is insufficient in comparison to its commitment. South Africa's total GHG emissions is 477 MtCO₂e, and 41% of these emissions are related to electricity production (USAID, 2015). Approximately 85.02% of the electricity in South Africa, supplied by Eskom, is obtained from coal (ESKOM Generation Division, 2021). This heavy reliance on coal as an energy source carries significant environmental impacts, primarily the emission of GHGs.

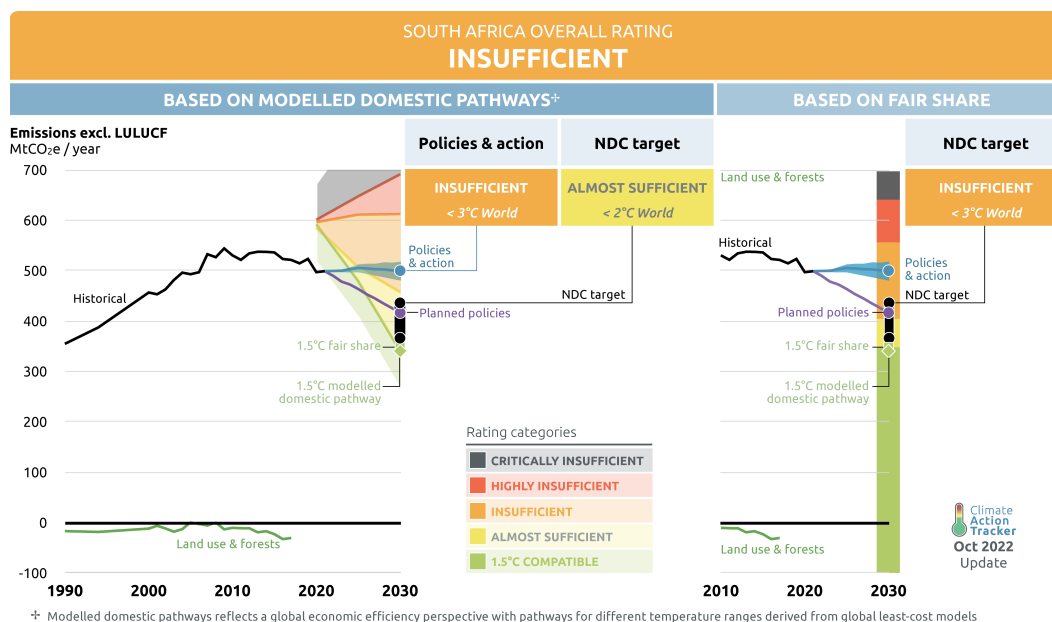


Figure 2: Summary of South Africa's Overall Current Rating (Climate Action Tracker, 2023).

To support the reduction in carbon emissions and promote a low-carbon economy, there are various measures that South Africa has established including the introduction of a Carbon Tax in 2019 (SA Department of Environmental Affairs, 2020). This policy includes the incorporation of an additional carbon offset allowance as a proactive measure to mitigate carbon emissions. Initially set at a rate of R120 per ton of CO₂e, the tax rate has been subject to annual adjustments based on the Consumer Price Index (CPI) plus 2%. In future years, it will be adjusted by CPI only. As a result of these adjustments, the current price valid for 2023 stands at R159/tCO₂e. Carbon offset allowances entitle these taxpayers to a reduction in the tax by a percentage of their emissions, which are obtained if they invest in projects aimed at reducing their carbon footprint. The implementation of the carbon tax follows a phased approach, and the initial phase is scheduled to conclude on 31 December 2025. As a result, taxpayers will continue to enjoy a tax-free allowance of 60% during this period.

Additionally, some of the planned policies that are yet to be implemented are the introduction of voluntary carbon credits under the carbon budget, subsidies, and Sectoral Emissions Targets (SETs) (SA Department of Environmental Affairs, 2020). Through this South Africa will aim to mitigate carbon emissions and promote sustainable practices.

South Africa contributes to 1% of the global emissions, which is a relatively small percentage of the total carbon footprint compared to countries like China or the USA (Mashishi, 2012). Hence, one may argue that mitigation policies are not relevant for the country. However, the incoming carbon border tax in the EU poses a concern. Considering that 22% of South Africa's exports are traded with the EU (European Commission, 2019), the carbon boarder tax would adversely affect this amount. The EU would likely divert its market to countries where robust tax policies are in place, in order to avoid paying the carbon border tax. Despite the country's relatively small contribution to global emissions, the implications of the carbon border tax present the importance of adjusting to international standards to mitigate the risk of export market disadvantages and maintain competitiveness.

2.3 Climate Change Risks (Transmission Channels)

The effects of climate change will necessitate significant structural changes to the world economy and financial system. The different transmission routes through which climate risks could affect the economy and financial system are depicted in Figure 3. Transition risks arise from the response, which is often from governments or policymakers to try and minimize climate change, as opposed to physical risks, which typically emerge from climate change directly Battiston et al. (2021).

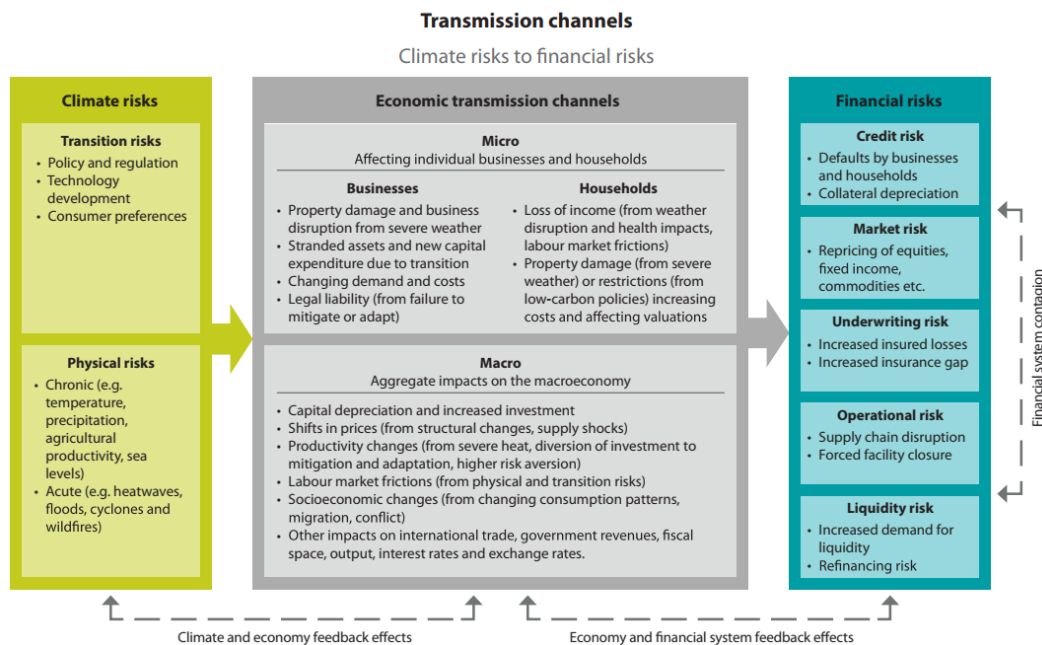


Figure 3: Climate risks transmission channels (NGFS, 2022).

1. Physical risks include the effects of both increasing temperatures and increasingly frequent extreme weather occurrences Battiston et al. (2021), NGFS (2022). The risks that arise are divided into two categories:
 - a. Acute risks, which are abrupt, severe, rapid events that have a large negative effect, such as flooding from heavy rain, which can cause significant disruption and property damage. These occurrences might increase insurers' underwriting risks, potentially resulting in less insurance coverage in some areas, and depreciating asset values.
 - b. Chronic risks are a result of steadily deteriorating ecological circumstances, such as rising sea levels and precipitation. In some locations, these could have an impact on labor, capital, land, and natural capital. Companies, households, and governments will need to make a large investment to adapt to these changes.
2. Transition risks are connected to promoting the shift from our existing forms of production to a climate-friendly economy. Risks for lenders and investors arise as a result of the effect this will have on individual wealth and business profitability NGFS (2022). Through investment, productivity, and relative pricing channels, they will also have an impact on the larger economy, especially if the transition results in stranded assets.

Recent research has presented several results on the inclusion of climate change-related physical and transition risks in the pricing of assets in financial markets. Extreme weather events like hurricanes, as an illustration of physical risk, make borrowing conditions worsen for sovereigns in the Caribbean Mallucci (2022). Bolton and Kacperczyk (2021) discover that, in the setting of transitional risk, stocks of firms with higher total CO₂e emissions (and variations in emissions) generate higher returns, which cannot be accounted for by differences in size, book-to-market ratio, or other return predictors. This implies that investors are seeking payment for their exposure to these carbon-intensive businesses. Generally speaking, these presented methodologies, provide information and in-depth analysis on these risks based on historical data, but they fall short of fully capturing the possible future effects of climate-related concerns. Therefore, the necessity to take into account upcoming uncertainties related to climate change is what motivates the inclusion of Network for Greening the Financial System (NGFS) scenarios in pricing both physical and transitional risks.

2.4 Network for Greening the Financial System (NGFS)-Scenarios

The implications of climate change are difficult to predict and pose enormous financial risks. Hence, NGFS climate scenarios are used to conduct in-depth analyses of these financial risks. Presently, there are six NGFS scenarios in three categories (NGFS, 2022).

1. **Orderly:** Early implementation of climate mitigation strategies is followed by a gradual tightening of restrictions. Physical and transition risks are both comparatively low. The related scenarios are Net zero 2050 and Below 2°C.
2. **Disorderly:** A greater transition risk: Policies take longer to implement or diverge between countries and industries, leading, for example, to higher carbon costs for a given temperature. The related scenarios are Divergent Net zero and Delayed transition
3. **Hot House world:** Despite some countries enforcing climate regulations, overall efforts to avert major global warming and its longer-term physical repercussions, such as extreme sea level rise, are insufficient. The related scenarios are Nationally Determined Contributions and Current Policies.

According to a coherent and internally consistent collection of assumptions regarding important driving forces (such as the rate of technological advance and carbon costs), these scenarios provide a credible depiction of how the future may unfold. It is vital to note that the given scenarios are used to provide a picture of the implications for future developments (neither the most probable nor desired) for financial risk assessment, rather than predictions or forecasts (NGFS, 2022). By setting up flexible-linear programming where a carbon net-zero constraint can be added and thereby introducing project redesigns that are financially net zero and thus achieving mitigation of carbon obligation, Kenyon et al. (2023), focusing on project finance, provides an example for incorporating these scenarios.

These six scenarios are employed in this paper to incorporate transition risk in the context of risk management (bank book) and derivative pricing (trading book), with a focus on the risk associated with the future evolution of carbon prices.

3 Model Assumptions

3.1 Assumptions of the Classical Merton

Merton's (1974) seminal paper has provided invaluable guidance on the use of structural models to model default risk. The Merton (1974) model offers insights into the effects of a firm's capital structure on its probability of default. Following the description of Wang (2009), the model assumes a simple capital structure, where at time $t \geq 0$ the firm has assets A_t financed by equity E_t and liabilities L_t , i.e.

$$A_t = E_t + L_t.$$

Mapping the debt into a zero-coupon bond with face value K at maturity T , the company may only default at maturity, which happens if $A_T < K$. Otherwise the

debt holders can be paid their full amount and the share holders' equity still has a value of $A_T - K$. Therefore, the equity value at maturity is

$$E_T = \max(A_T - K, 0),$$

which can be interpreted as the payoff of a call option on the firm's assets with strike K . Under the assumption that the value of the assets follows a Black–Scholes model,

$$dA_t = rA_t dt + \sigma A_t dW_t,$$

where W_t is a Wiener process, $r \in \mathbb{R}$ is the interest rate and $\sigma > 0$ is the volatility, we can compute probabilities of default explicitly. In particular, according to the Black–Scholes formula for European call option pricing, we have

$$E_t = A_t \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2),$$

where Φ is the standard normal CDF,

$$d_1 = \frac{\ln(A_t/K) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma \sqrt{T-t}},$$

and

$$d_2 = d_1 - \sigma \sqrt{T-t}.$$

Hence, under the risk-neutral measure, the probability of default is

$$Pr[A_T < K] = \Phi(-d_2).$$

3.2 Assets and Liabilities Model

In contrast to the simplifying assumption in (Merton, 1974) that the assets follow a Black–Scholes model, we consider the general structural equity model

$$E_t = A_t - L_t,$$

where we model the main income and cost streams of a power utility firm by the assets A_t and liabilities L_t , respectively. For simplicity, we consider a discrete time setting, with times $t \in \{0, 1, \dots, T\}$. Then, the evolution of assets and liabilities (both stated in South African Rand (ZAR)) through time are given by

$$A_t = A_0 + \sum_{s=0}^t C_s \times E_s \times \delta_s \quad (1)$$

and

$$L_t = L_0 + \sum_{s=0}^t (e_s + F_s + DEP_s + (R_s + M_s) + NET_s) \cdot \delta_s, \quad (2)$$

where we consider the processes discounted to time $t_0 = 0$ via the discounting factor δ_t . The asset value changes with time t according to the power utility firm's capacity C_t (in MW) and the prevailing electricity price E_t (in ZAR/KWh), which is set by NERSA National Energy Regulator of South Africa (2023). The initial assets A_0 and liabilities L_0 are obtained from the balance sheet for the current year at the starting point $t = 0$. The liabilities L_t arise from the total costs (in ZAR) incurred from costs of carbon emissions e_t , depreciation costs DEP_t , fuel's cost F_t , maintenance costs M_t and running costs R_t and the Negative emissions technology (NET)-investments NET_t , all discounted to t_0 . (NET)-investments are relate to making investments in technology aimed at lowering atmospheric CO_{2e} concentrations, either through deliberate enhancement of land and ocean carbon sinks to speed up the removal of CO_{2e} from the atmosphere or through engineering its removal and subsequent storage.

In this study, the stochastic process for the electricity price is defined as

$$E_t = E_{t-1} \times X_t = E_0 \times \prod_{s=1}^t X_s, \quad (3)$$

Where $(X_s - 1)$ are identical and independent random variables drawn from an exponential distribution, i.e., $(X_s - 1) \sim \text{EXP}(\lambda^{-1})$, where λ represents the mean percentage price jumps (e.g., 0.2 represents a 20% average increase).

3.3 Minimum Electricity Price to Cover the Total Costs

Given that all other quantities are known, it is possible to determine at every time step t the electricity price $E_t^{\min}$ such that the costs to produce the electricity are exactly covered by the income of selling the electricity by solving

$$A_t - A_{t-1} = L_t - L_{t-1} \quad (4)$$

for E_t . In particular, this yields

$$E_t^{\min} = \frac{e_t + F_t + DEP_t + R_t + M_t + NET_t}{C_t} \quad (5)$$

Computing this minimal electricity price for different NGFS scenarios allows us to compare the costs resulting from emissions given the fuel mix of the power utility company.

3.4 Probability of Default and Running Probability of Default

The default probability of the power utility firm at time $t \geq s$ conditioned on the information available at s , $\mathcal{F}_s$, is given by

$$\tilde{P}(s, t) := \mathbb{P}(A_t < L_t) = \mathbb{E}[\mathbb{I}_{\{A_t < L_t\}} | \mathcal{F}_s]. \quad (6)$$

Table 1: Eskom’s balance sheet data as of March 31, 2022. NB: BZAR = Billion ZAR, MT = million tons, TWh = terawatt hours.

Equity	235.314 BZAR
Annual Depreciation Costs	32.009 BZAR
Annual Maintenance Costs	24.113 BZAR
Annual Labour Costs	32.985 BZAR
Annual Sold Energy	192.005 TWh
Annual CO ₂ e Emission	207.230 MT

Moreover, the running probability of default is defined as

$$P(s, t) := \mathbb{P}(\exists k \leq t : A_k < L_k | \mathcal{F}_s). \quad (7)$$

Hence, the price of the corresponding defaultable bond is

$$B(s, t) = \delta_t \cdot \mathbb{P}(\forall k \leq t : A_k \geq L_k | \mathcal{F}_s). \quad (8)$$

We note that $\tilde{P}(s, t)$ is the probability that the firm defaults at time step t , while $P(s, t)$ is the probability that the firm defaults at any time up to time t . In the following we mainly consider the running probabilities of default and their corresponding bond prices, both conditioned on the initial time t_0 . In particular, we consider their term structures $t \mapsto P(0, t)$ and $Bt \mapsto (0, t)$ for $0 < t \leq T$.

4 Model Example

4.1 Eskom Overview

In this paper we consider the primary energy supplier in South Africa, which is the state-owned company ESKOM SOC Ltd (Eskom) supplying the majority of electricity in South Africa. It is accountable for both electricity generation and distribution, serving industries and municipalities across the country. Furthermore, Eskom also buys electricity from Independent Power Producers (IPP) and international sources based in southern Africa (ESKOM, 2022). In Table 1 we show Eskom’s financial data relevant to us, which was published in their latest financial statement (ESKOM, 2022). Moreover, in Table 2 we show the combination of power plant types Eskom uses to generate electricity. Diesel and gas turbines are used during periods of sudden demand increase or peak times, as they have high operation costs.

The historical tariffs of Eskom have shown a gradual increase over the years. Table 3 shows the tariff trend, spanning a 10 year time period, from 2012 to 2022.

Table 2: Eskom mix of electricity generating plants

Source	Capacity (MW)	Ratio
Coal	44013	85.02%
Nuclear	1934	3.74%
Renewable	3393.4	6.56%
Gas	2426.3	4.69%
Total	51766.7	100%

Table 3: Eskom electricity sell price tariffs from 2012 - 2022

Year	Tariff (ZAR/kWh)	Percentage change
2012	0.5849	-
2013	0.6281	7.3859
2014	0.6763	7.6739
2015	0.7538	11.4594
2016	0.8177	8.4770
2017	0.8249	0.8805
2018	0.8512	3.1883
2019	0.9001	5.7448
2020	1.0186	13.1652
2021	1.1104	9.0124
2022	1.2732	14.6614
Total % increase		117.6782

Eskom introduced load shedding as a measure which safeguards the grid supply from unplanned blackouts through a controlled process of shutting down the electricity supply in certain regions for several hours (Matsheta and Sefoka, 2023). This deliberate action was taken by Eskom when there is inadequate capacity to meet the demand. Figure 4 shows how load shedding has increased due to a decrease in realised electricity output over time. Consequently, these shifts have resulted in lower carbon emissions.

In terms of financing, Eskom raises funds by issuing debt in both domestic and international debt capital markets. Eskom issues domestic bonds under the Domestic Multi Term Note (DMTN) programme, enabling them to be traded under the Johannesburg Stock Exchange (JSE). Part of the domestic bonds are guaranteed by the government of South Africa. On the other hand, international Eskom bonds are US Dollar denominated registered under the Global Medium-Term Note (GMTN) programme. This programme is registered under the Luxembourg Stock Exchange (LSE), and the issuance of the international bonds is facilitated by the

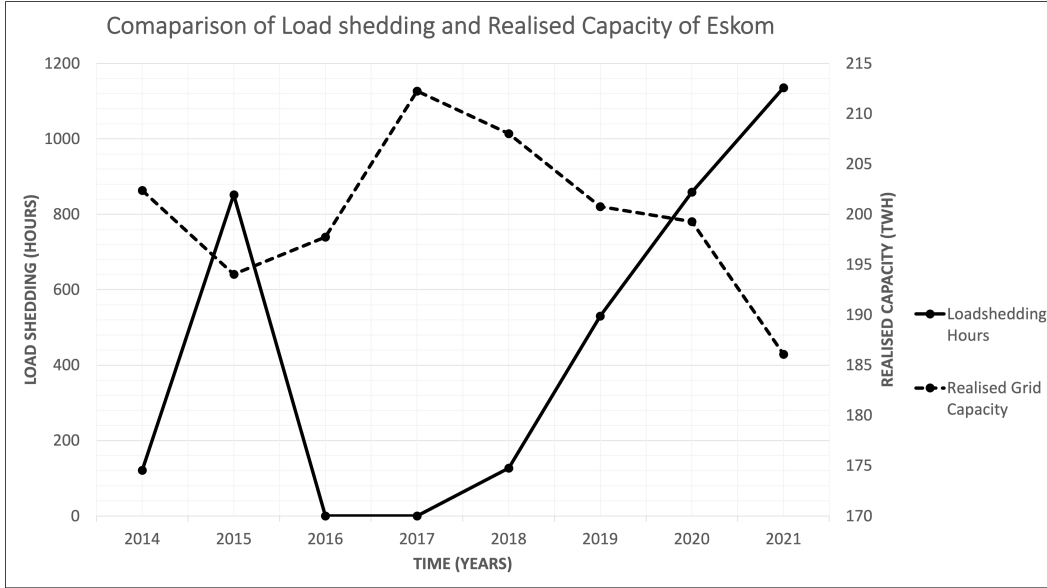


Figure 4: Eskom realised electricity capacity vs annual load shedding from 2014 to 2021.

banks that are authorised by Eskom. Both these bonds are available for purchase by investors.

4.2 Data Description

The study makes use of bond price data and future fuel prices (coal, natural gas, etc.) from Bloomberg. In particular, we use the South African Government Bonds (SAGB) and Eskom bonds as well as US government bonds. The South African inflation and discount factors are derived from the inflation-linked and nominal SAGBs, respectively. The US inflation and discount factors are derived equivalently.

The (market) probabilities of default of Eskom are derived from Bloomberg. Moreover, we use Bloomberg data to derive the probabilities of default corresponding to different credit ratings, for an easier comparison. In particular, we use the 5Y, 10Y and 20Y yield curves corresponding to S&P's credit ratings "AAA" to "B-" to obtain their credit spreads. The difference between this spread and the corresponding US government bond is used to determine their default intensity (λ_T), extrapolated

to any maturity T via

$$\lambda_T = \begin{cases} \lambda_5, & 0 \leq T < 10, \\ \lambda_{10}, & 10 \leq T < 20, \\ \lambda_{20}, & 20 \leq T. \end{cases}$$

These values were then used to compute the probability of default corresponding to the credit ratings

$$P^D(T) = \frac{1 - e^{-\lambda_T T}}{(1 - R)}, \quad (9)$$

and R is the recovery rate assumed to be the market standard of 0.4.

We take the monthly prices for all the fuel futures (where available, otherwise the current price kept constant afterwards), average them over a year, and then maintain the price from the most recently observed price until 2050. To get a reasonable price evolution, these prices are adjusted for US inflation and converted to ZAR by the FX-rate of 16.9891 ZAR/USD on January 1, 2023. Moreover, they are discounted with the US interest rate whenever discounting is applied to fuel costs in our model (equivalently, they are multiplied by the US discounting factor and divided by South African one, which leads to the correct discounting in the model when using the standard South African discounting factor). The reason for using US rates instead of South African ones is that the commodity market is US dollar based. It is important to mention that Eskom gets a special coal price from the local South African coal mining companies National Energy Regulator of South Africa (2023). In particular, it only pays 44.3% of the market coal price. In our model, we adjust the coal price accordingly.

The depreciation costs (in ZAR/KWh) per year displayed in Table 4 are computed as

$$DEP_t = \frac{\text{Capital costs/life span}}{\text{Capacity} \cdot 365 \cdot 24 \cdot \text{capacity factor}}. \quad (10)$$

Table 4 shows the capital costs, lifespans, capacities, and capacity factors, as presented in Kenyon et al. (2023), which are used to compute the depreciation costs for the different power plant types. We note that these numbers are “prototype” power plants and therefore might need to be adjusted for the actually installed power plants.

The South African Integrated Resource Plan (IRP) of 2019 IRP (2019), which was passed by the South African government, describes the planned energy additions (with a drift towards more renewable fuel types, especially wind) and the decommissioning of coal capacities until 2030. The changes relevant to our model are displayed in Table 5. We infer Eskom’s capacities of the different fuel types in the

Table 4: Technology characteristic, financial costs and life span. NB: Carbon Capture and Storage (CCS), BZAR = Billion South African Rands, MW = Mega Watts.

Technology	Size (MW)	Capital cost (BZAR)	Life span (Years)	Capacity factor	Depreciation costs (ZAR/KWh) per year
Ultra-supercritical coal (USC)	650	43.356	40	0.85	0.223
USC with 90% CCS	650	69.299	40	0.85	0.358
Combined-cycle-single shaft	418	8.223	40	0.87	0.0645
Combined-cycle with 90% CCS	377	16.972	40	0.87	0.148
Nuclear-small modular reactor	600	67.395	40	0.9	0.356
Conventional hydropower	100	9.650	50	0.5	0.441
Wind onshore	200	4.587	25	0.38	0.276
Wind offshore	400	31.753	25	0.39	0.929
Solar photovoltaic (PV) with tracking	150	3.568	30	0.158	0.573

Table 5: Decommissioning of existing capacity and addition of new capacities between 2023 and 2030 IRP (2019)

Year	Addition-Coal (MW)	Coal-Decommission (MW)	Change in Coal (MW)	Addition-Nuclear (MW)	Addition-Hydro (MW)	Addition-Solar (MW)	Addition-Wind (MW)	Addition-Gas and Diesel (MW)
2023	750	-555	195	0	0	1000	1600	0
2024	0	0	0	1860	0	0	1600	1000
2025	0	0	0	0	0	1000	1600	0
2026	0	-1219	-1219	0	0	0	1600	0
2027	750	-847	-97	0	0	0	1600	2000
2028	0	-475	-475	0	0	1000	1600	0
2029	0	-1694	-1694	0	0	1000	1600	0
2030	0	-1050	-1050	0	2500	1000	1600	0

years until 2030 from the IRP 2019.

The price evolution of CO₂e from 2023 to 2050 (in ZAR/kg) for the six NGFS scenarios and the South African carbon tax (cf. Section 2.2) are shown in Figure 5. The "No cost" path assumes that the price of CO₂e remains constant at zero. The carbon prices data for the six NGFS scenarios are obtained from NGFS Portal. The prices from the scenarios are already US inflation adjusted and given in 2010 USD. Therefore, we adjust them by inflation for January 1, 2023 (which is our initial date) and convert them to ZAR (as for the fuel prices described above). Moreover, in our model we discount future carbon prices by the US discounting factor (used for the same reason as for the fuel prices) to the initial date. Under the most extreme scenario, the Divergent Net Zero path, the (non-discounted) CO₂e price is already slightly higher than 2.7 ZAR/kg in 2023 and rises significantly to more than 80 ZAR/kg until 2050. Compared to this, the emission price in the South African carbon tax scenario, which is very similar to the current policies scenario, only increases to about 1 ZAR/kg and in all other scenarios, the price lies in between these

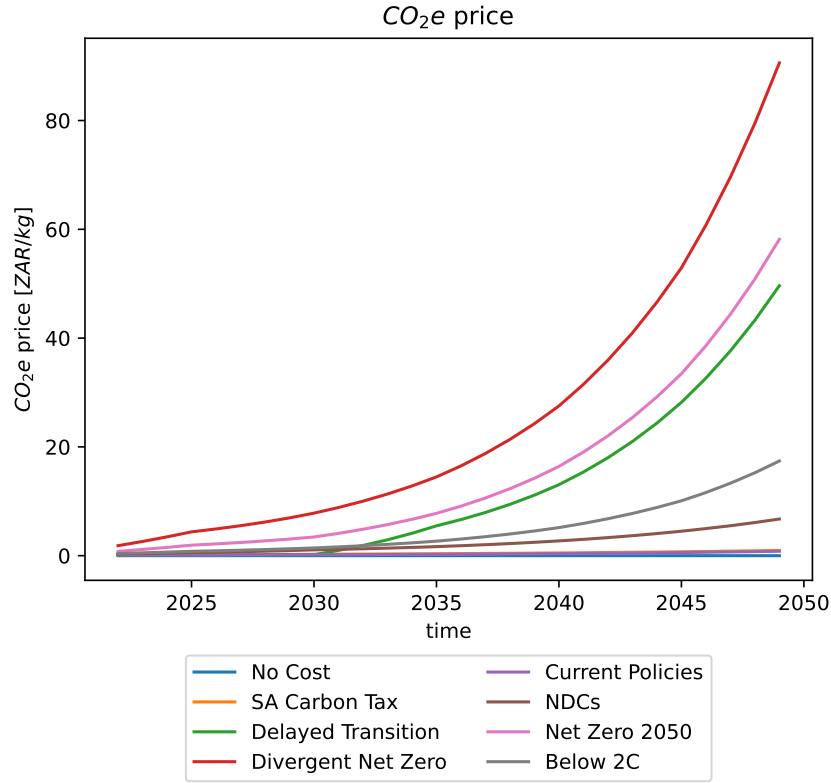


Figure 5: CO₂e price according to South African carbon tax and NGFS Scenarios.

extremes.

We can now combine the data on fuel costs, depreciation costs and emission costs to visualize the cost trend (excluding running costs) from 2023 to 2050, as shown in Figure 6 and 7. It should be noted that "clean coal" refers to USC with 90% CCS, while "coal" refers to USC and similarly "clean gas" refers to Combined-cycle with 90% CCS, while "gas" refers to Combined-cycle-single shaft (cf. 4). The cost of nuclear, solar, hydro, and wind energy never exceeds 1.0 ZAR/KWh in any of the scenarios, indicating that investing in clean energy technology is a financially sound decision. Coal appears to be fairly cost-competitive with clean technologies under the no cost, South African Carbon Tax, current policies, and NDC's scenarios (also due to Eskom's special coal price). However, in other scenarios, we notice an increase in the cost of coal-based technologies (even exceeding 4 ZAR/KWh for Divergent Net Zero path in 2050), which reflects higher expenses caused by the CO₂e emissions. Coal plants with carbon capture and storage are a financially more effective alternative. In all of the scenarios, the costs for gas and clean gas are observed to range from 2.2 to 5 ZAR/KWh and the costs for diesel are the highest with 7.8 to

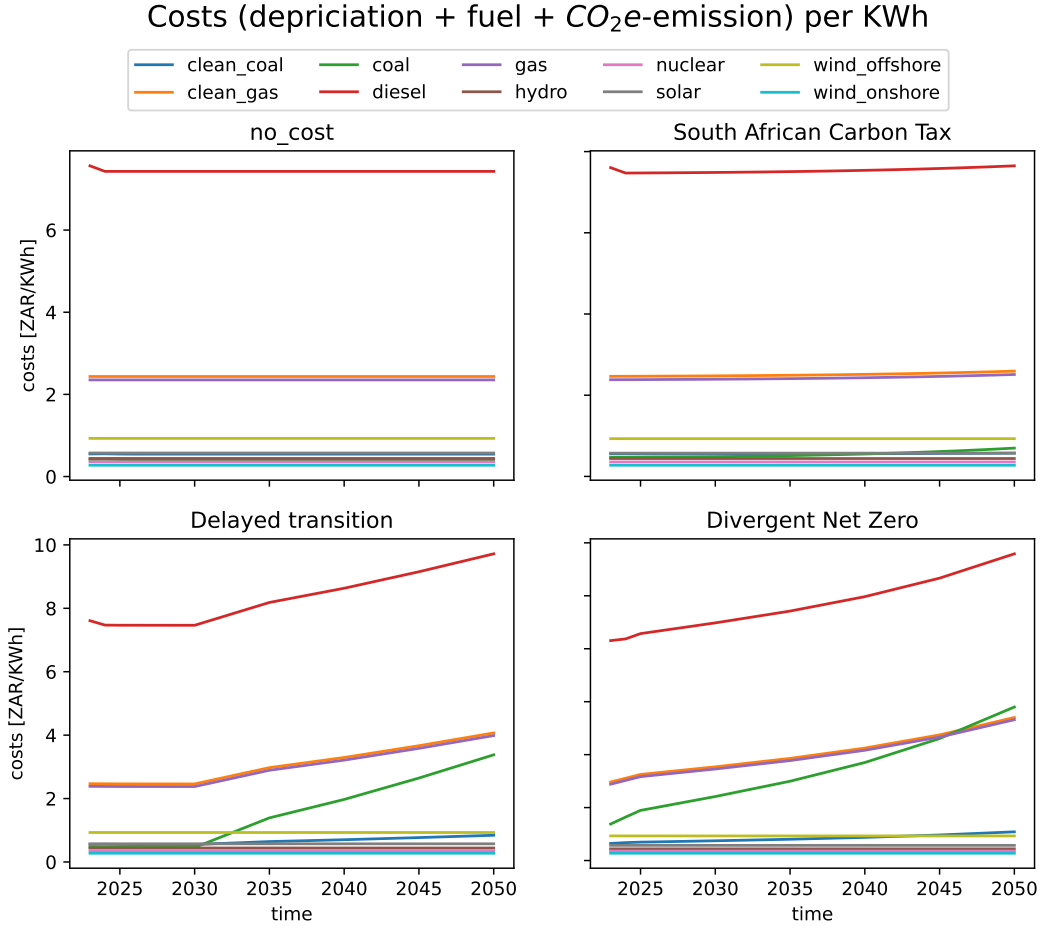


Figure 6: Costs (depreciation, fuel and CO₂e emissions) for different power plant types for No cost, South African Carbon Tax, Delayed Transition and Divergent Net Zero.

10 ZAR/KWh. Thus, considering generation technologies based on diesel would not be financially viable.

5 Results and Discussion

5.1 Parameters and Considered Scenarios

In this section we discuss the different parameters and assumptions for the proposed model.

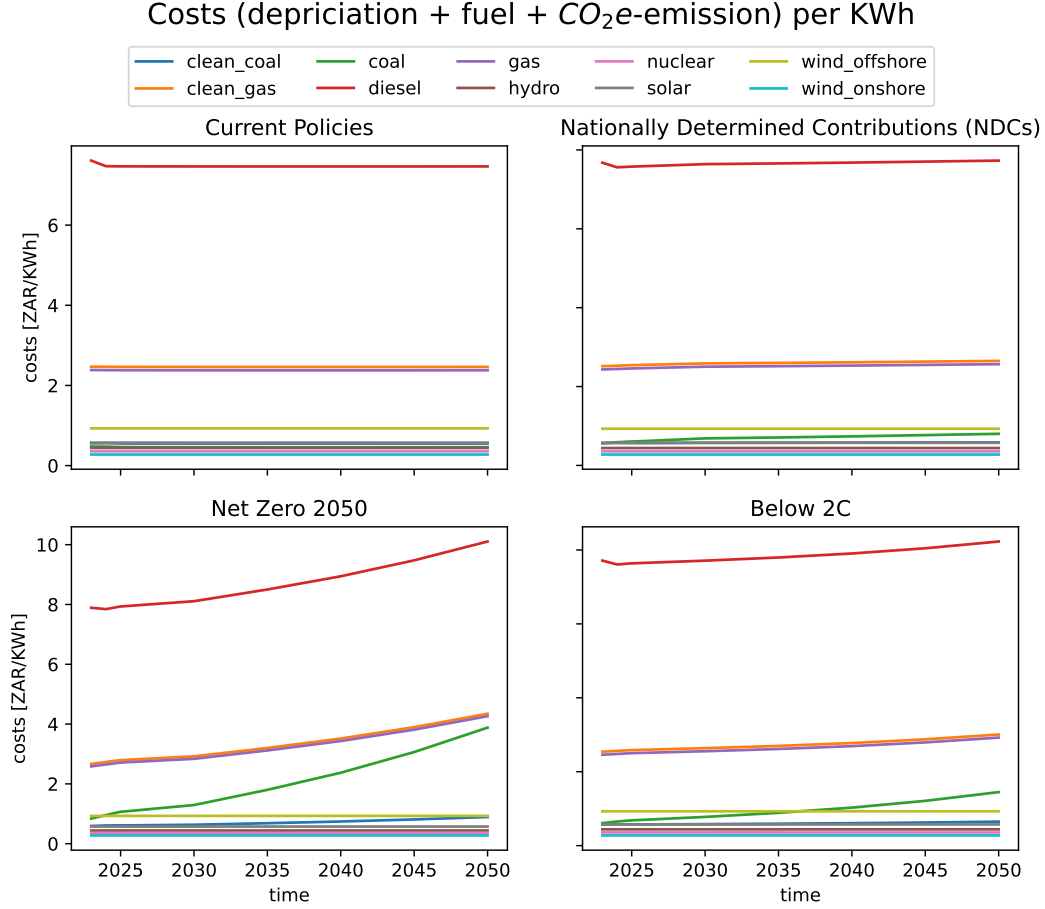


Figure 7: Costs (depreciation, fuel and CO₂e emissions) for different power plant types for Current Polices, Nationally Determined Contributions (NDCs), Net Zero 2050 and Below 2°C.

5.1.1 Base Case: IRP 2019

The base case that we consider represents our assumption about the scenarios that are currently assumed in and therefore priced by the market. Hence, we start by explaining these scenario and parameter choices. The start date t_0 of our model is January 1, 2023 and we use yearly time steps until January 1, 2051 corresponding to a maturity of 28 years.

Starting from the initial capacity of Eskom, as stated in Table 2, this base case assumes that the capacities of the different plant types are changed until 2030 as stated in the IRP 2019 of South Africa (cf. Table 5). More precisely, we make the conservative assumption that the changes stated there are available by the begin-

ning of the next year (e.g. changes stated for 2023 are assumed to be in place as of 2024 but not earlier). After 2030, the capacities are assumed to be constant until maturity.

In each year, the total maximal capacity (i.e. the capacity stated in Table 2 and Table 5) is adjusted using the capacity factors outlined in Table 4 for the different power plant types, in order to determine the total (theoretically) realisable capacity. In reality only a fraction, denoted as the capacity production factor β_{cp} , of the theoretically realisable capacity is actually used (on average) to produce the sold energy within one time step. Comparing the realisable electricity, i.e. the electricity when running all available plants on their theoretically realizable capacity for the entire time period, to the sold electricity (cf. Table 1), we can compute $\beta_{cp} = 51\%$. Therefore, we adjust the total realisable capacity in every year accordingly (keeping the capacity production factor constant over time and applying it to every power plant type equally) to get the average production capacity C_t .

Since Eskom’s financial statement (ESKOM, 2022) does not state any NET investments (carbon absorption / negative emissions) they are set to zero in the model. The yearly running costs, which encompass maintenance costs and labour, is defined to be the respective values published on Eskom’s balance sheet in 2022 (cf. Table 1) scaled by C_t/C_1 for the following years.

We assume that the market uses the South African carbon tax scenario to price Eskom at the moment. Therefore, we also use this carbon price when calibrating our model to the market data of the probabilities of default.

The electricity price is assumed to follow our stochastic price model (3), which experiences an annual random increase. The parameter λ of the yearly mean percentage price jump is calibrated to the market. Additionally, we calibrate α to the market, where α determines how much of the initial equity (stated in Table 1) is used in the model. The reason for using a fraction of Eskom’s stated equity is that our simplistic model does not capture more complex relationships (e.g. interest rate payments that Eskom has to make and additional costs as for example running the power grid). This factor allows to adjust for these simplifications. In particular, we use the 2 degrees of freedom (λ, α) to calibrate our base-case model described in this section to the market default probabilities. By doing this, we assume that the described parameters (i.e., the IRP 2019 capacity changes and the South African carbon tax) are the most plausible to be used for pricing by the market.

The calibration of the model is done by choosing the best parameter combination $(\lambda, \alpha) \in \{(\frac{i}{99}, \frac{j}{19}) | 0 \leq i \leq 99, 0 \leq j \leq 19\}$. In particular, for each parameter combination we compute the running probabilities of default (7) with our model using 10,000 i.i.d. sampled electricity paths. Then the combination (λ^*, α^*) that min-

imises the 2-norm between our model implied probabilities of default and probabilities of default for Eskom observed at the market is chosen. In Figure 8 we show the model default probabilities for the optimal parameters $(\lambda^*, \alpha^*) \approx (0.07, 0.16)$ together with the market default probabilities. We observe, that the model fits the market data relatively well, even though we only have 2 degrees of freedom for fitting the model.

We note that all results shown in Section 5.2, use the same 10,000 randomly sampled electricity price paths to make results well comparable.

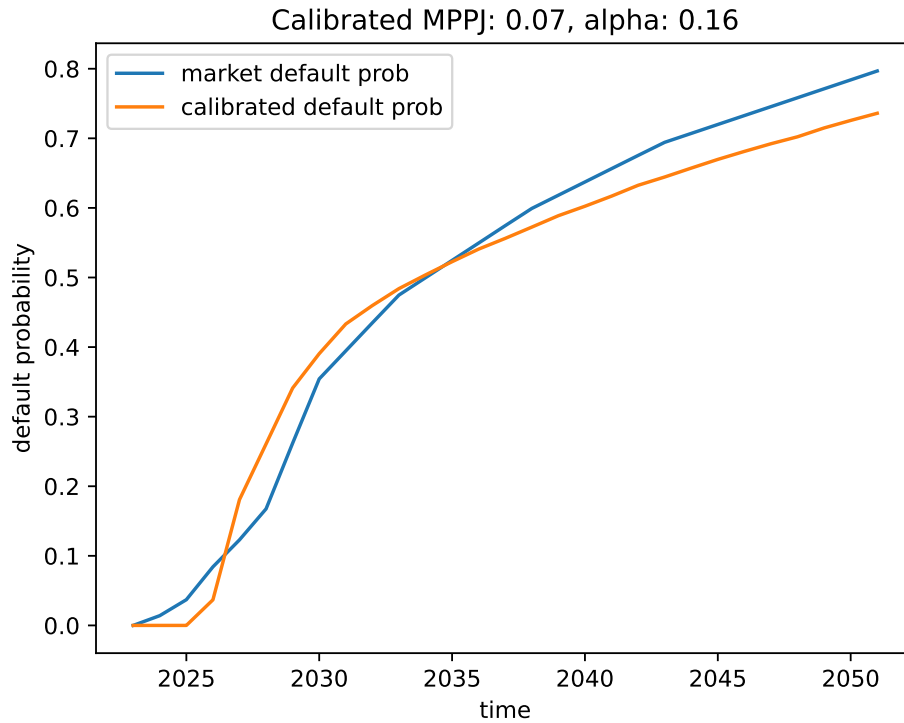


Figure 8: Calibrated Mean Percentage Price Jump.

5.1.2 Second Case: IRP-2019 with "Green Continuation"

In the second case, all the parameters obtained from the base case are reused, except for the capacities. In particular, we use the mean percentage price jump (MPPJ) and the initial equity fraction (λ, α) that we obtain in our calibration (cf. Figure 8). The difference to the base case is that the capacities for the different power plant types are not assumed to be constant after 2030 but are continuously changing until maturity as follows. The coal capacity is continuously decreased after 2030

by -1037MW per year, while the solar capacity is increased by 1000MW and the wind onshore is increased by 1600MW per year (these changes correspond to their respective average changes in the last 3 years of the IRP 2019). Moreover, every 10 years, the nuclear energy capacity is increased by 1860MW and the hydropower capacity is increased by 2500MW , corresponding to the changes over the entire period of the IRP 2019).

5.1.3 Third Case: IRP-2019 with “Aggressive Green Continuation”

In the third case, we again use the same parameters as in the base case and only change the capacities. In particular, the capacity changes for coal and solar energy are doubled from the second case to -2000MW and 2000MW , respectively. An annual increase of 1600MW is considered for the onshore wind and an additional yearly increase of offshore wind by 1600MW is assumed. The hydropower capacity change is not considered to be increasing once every 10 years as in the second case, but to have a yearly increase of 1000MW . The nuclear capacity is increased every 10 years by 1860MW , identical to the second case.

5.1.4 Fourth Case: IRP-2019 with “Green Continuation with higher Electricity Prices”

The parameters in this case are assumed to be the same as in the second case of green continuation. The difference between the two cases is that a MPPJ of 10% was used instead of the 0.07% yielded by the model calibration. We can see in the left plot of Figure 9 that when the calibrated MPPJ is used to calculate the electricity prices until maturity, the inflation adjusted price starts to slightly decrease after 2030. Therefore, it is arguably more realistic that the actual prices grow faster. A mean price increase by 10% is in accordance with Eskom’s plan outlined in the Annual Financial Statements (ESKOM, 2022) and very close to the average price increase within the last 10 years (cf. Table 3).

5.2 Discussion

5.2.1 Capacities Energy Mix

Figure 10 presents the evolution of the capacity C_t and the realisable energy mix (i.e, the proportions of the different power plant types of the total capacity) in the 4 cases defined in Section 5.1. In the base case capacities and energy mix stay constant after 2030. In the 2nd and 4th case we assume that coal capacities decline gradually with the observed decommissioning trend from 5 while renewable and other capacities increase with the same trend. This leads to coal capacities contributing to about 30% in the overall mix by 2050, being similar to the final onshore wind contribution. In the 3rd case, we assume that coal decreases heavily to about

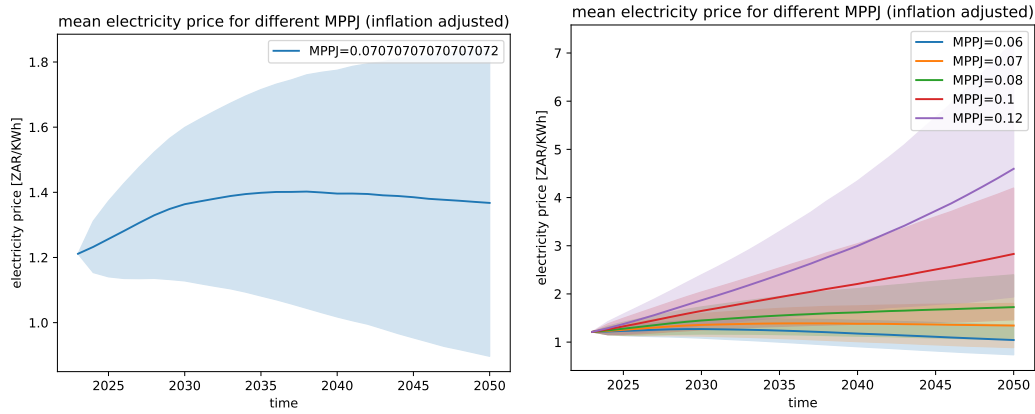


Figure 9: Mean $\pm$ standard deviation of electricity prices for different MPPJ parameters. Left: electricity prices from calibrated MPPJ parameter. Right: electricity prices for different MPPJ parameters.

2.3% in 2050, while renewable capacity penetrations increase heavily, especially onshore wind, to about 30% of the energy mix.

5.2.2 Running Default Probabilities for Different Cases.

Figure 11 shows the results for the running default probabilities matched with the CO₂e price for all the presented cases. Their corresponding bond prices are shown in Figure 12. It can be seen that in all cases, under the Divergent Net zero scenario, the running default probability already is 1 (giving it a rating B- or worse) before 2025 and the corresponding bond price is 0. This can be attributed to the huge costs that Eskom will incur due to escalating CO₂e prices. Considering the path evolution under the South African Tax with higher electricity price jumps (which is quite matched with the Current Policies scenario), it can be seen in Figure 11 (4th case) that the running probability of default increases slowly from 0 right after 2025 to slightly above 0.2 and hence reflecting a rating between (AA and AAA). Besides the assumed larger electricity price jumps this is due to a smaller proportion of coal in the energy mix. In the other two cases (2nd and 3rd) under the South African Carbon Tax and Current Policies scenario, the bond rating stays between BB and BBB with a running default probability above 0.6. However, we clearly see an improvement of the default probabilities by several percent from the base case to the 2nd and again from the 2nd to the 3rd case under all but the two most extreme carbon price scenarios (Divergent Net Zero and Net Zero 2050). In those two scenarios the initial electricity price and its growth are too small to cover the costs in the beginning, leading to certain default. In the 4th case we see, however, that a slightly increased mean percentage price jump of 10% for the electricity price reduces the other default probabilities drastically.

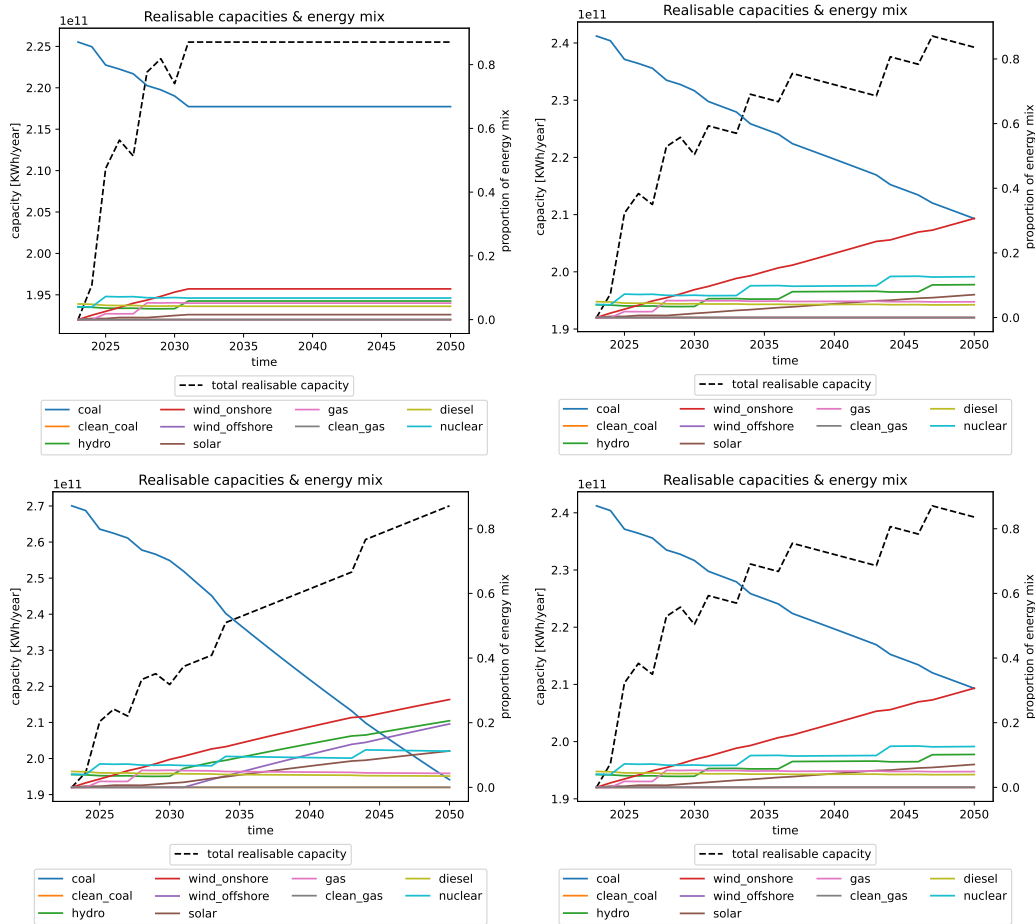


Figure 10: The capacities for all 4 cases. Top left: base case. Top right: second case. Bottom left: third case. Bottom right: fourth case.

5.2.3 Minimum Electricity Prices to Cover Costs

Figure 13 shows the minimal electricity price Eskom must charge to consumers in order to exactly cover all expenses in each time period. It is noticeable that there is a constant increase in electricity prices over time in the 1st, 2nd and 4th case, which is consistent with the different carbon emission price scenarios. The high costs associated with using CO₂e emission-based technologies, such as coal and gas, explain this behaviour. Due to the aggressive introduction of more renewable energy sources (particularly wind) in the 3rd case paired with the progressive eradication of coal, we observe steady increases from little over 2 ZAR/KWh in 2023 to around 3 ZAR/KWh between 2030 and 2035 in the Divergent Zero scenario. This is because there is still a significantly bigger share of coal in the energy mix, and

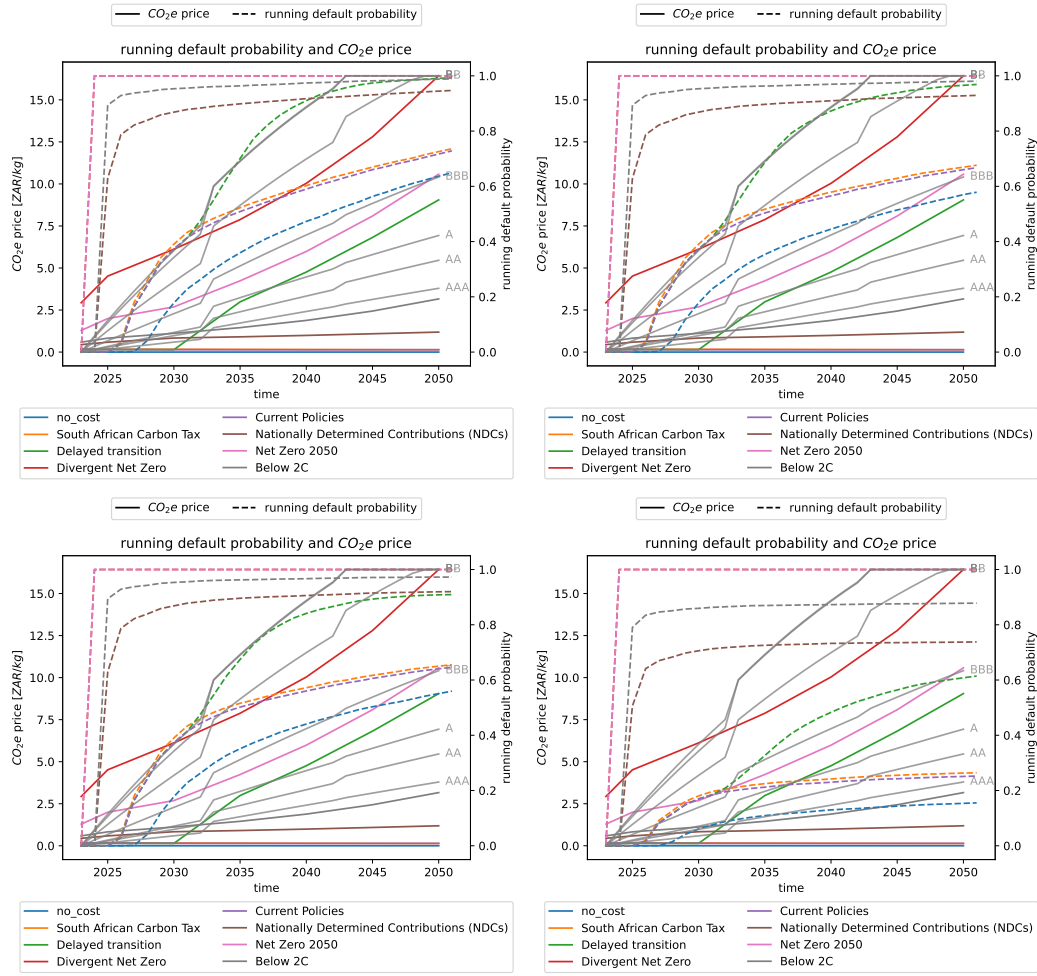


Figure 11: Running Default Probabilities. Top left: base case. Top right: second case. Bottom left: third case. Bottom right: fourth case.

as we add more renewable energy sources to the mix, starting in 2035, the minimal price of electricity to cover the costs starts to decline, reaching about 2 ZAR/KWh in 2050. In particular, there is a tendency that is consistent with all higher emission cost scenarios, namely that prices are increasing first, but start to fall steadily after 2045. As a result, this additionally encourages the need to invest more in renewable technology because Eskom will be able to sell power at an affordable price while also recovering costs because CO₂e emissions are less of an expenditure.

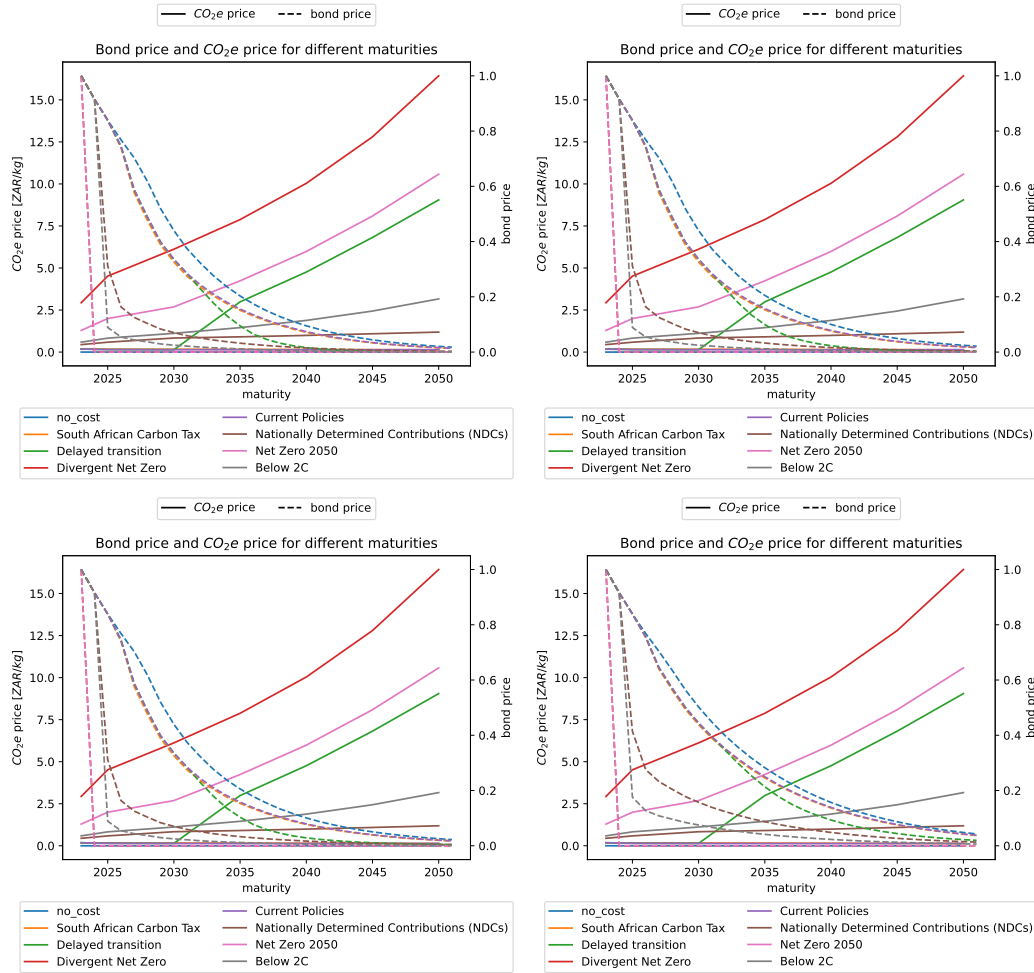


Figure 12: Bond prices for all cases. Top left: base case. Top right: second case. Bottom left: third case. Bottom right: fourth case.

6 Valuation adjustment

Up until now, the primary focus has been on the banking book. That is, the bank holds bonds of the credit-risky counterparty, Eskom. For the banking book, the bank typically holds assets until maturity, therefore, the main focus is the risk management of the book. This means that the bank needs to fully understand the risk of the assets in its portfolio and how this risk is changing for effective risk management. Moreover, it is crucial that the bank understands the risk dynamics so that it makes informed decisions about the instruments it adds to its portfolio. Equally important is the trading book which has not been broached. The bank would have derivatives and other financial instruments that are held for trading

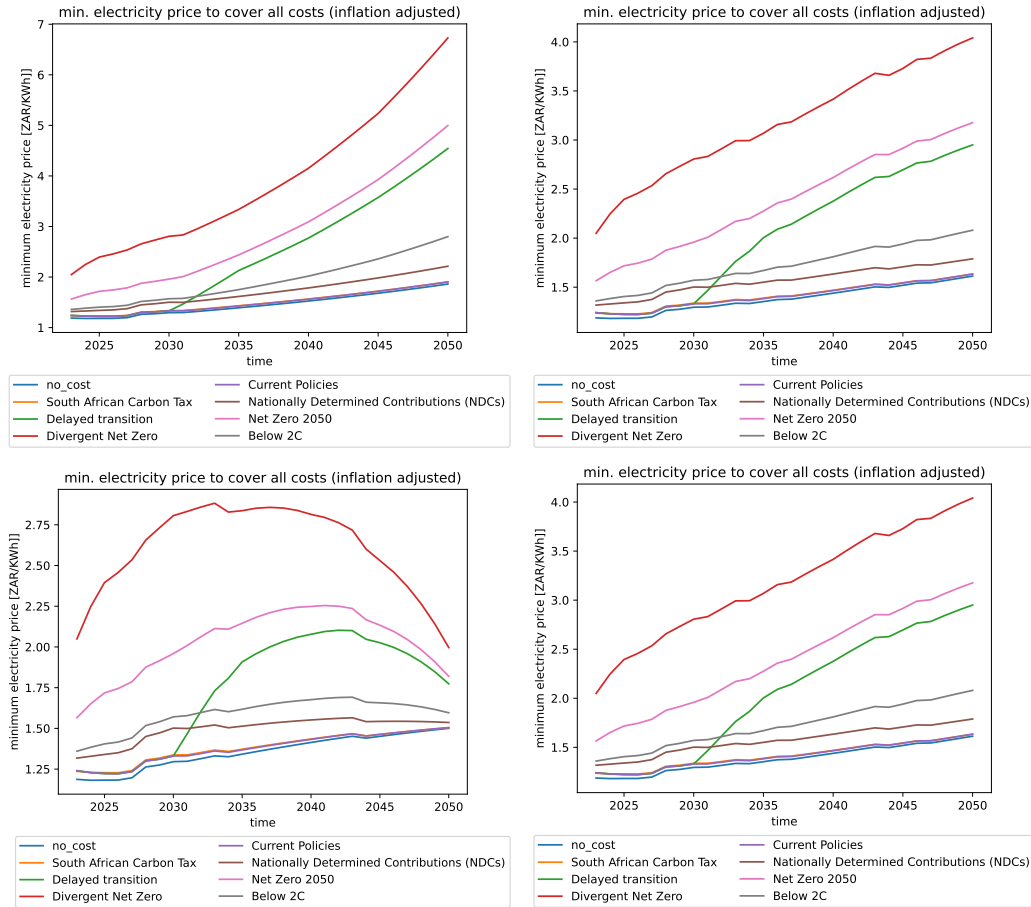


Figure 13: Minimum electricity prices to recover costs for all cases. Top left: base case. Top right: second case. Bottom left: third case. Bottom right: fourth case.

purposes. These instruments are typically liquid with active secondary markets and are marked to market daily in the trading book. The trading book is the focus of this section, more specifically, the pricing and hedging of the derivatives. This is imperative because the bank needs to completely understand the risk that comes with its positions, not only to avoid huge unexpected losses, but also to assess the capital charge arising from the risk of the positions and adhere to *Basel III* capital requirements.

6.1 Pricing interest rate swaps

This paper considers an example where the bank has a position in a vanilla interest rate swap (IRS). An IRS is a derivative contract in which two parties exchange interest rate cashflows based on a defined notional amount. Assuming that the

bank is the payer, it pays fixed coupons to its counterparty and receives floating payments which reference a short-term interest rate. The fixed rate is determined at the inception of the contract and remains unchanged until maturity. The set of times $\{t_0, t_1, \dots, t_{n-1}\}$ are called the reset dates and $\{t_1, t_2, \dots, t_n\}$ are called the payment dates because the reference rate resets at each of the former times and the payments are made at each of the latter dates. The set of default free discount factors observed at time t is given as: $\{Z(t, t_1), Z(t, t_2), \dots, Z(t, t_n)\}$ and the forward rates are given by $\{f(t; t_1, t_2), f(t; t_2, t_3), \dots, f(t; t_{n-1}, t_n)\}$. This paper assumes that the coupon accrual periods denoted by δ_i where $\delta_i = t_i - t_{i-1}$ for $i \in \{1, \dots, n\}$ are the same as the terms of the reference rates.

Under the no-arbitrage assumption and no default risk, the value of the swap at time t is given by

$$V(t; t_0, t_n) = \alpha \sum_{i=1}^n (f(t; t_{i-1}, t_i) - K(t_0, t_n)) \delta_i Z(t, t_i), \quad (11)$$

where α equals 1 (-1) for a long (short) position, $t_0 \geq 0$ is the effective time, t_n is the maturity time and $K(t_0, t_n)$ is the fair strike rate at time t_0 . More formally, the fair strike rate $K(t_0, t_n)$ is given by

$$K(t_0, t_n) = \frac{1 - Z(t_0, t_n)}{\sum_{i=1}^n \delta_i Z(t_0, t_i)} \quad (12)$$

The value of the swap to the bank (long party) at time $s > t$ will increase if the market strike rate increases, that is, the strike rate, $K(s, t_n) > K(t_0, t_n)$.

6.2 Credit Valuation Adjustment

Up to this point, the risk of default has not been considered in pricing the swap. If the bank's counterparty is susceptible to default risk, then this risk should be reflected in the price of the swap. This means that the price of the derivative must be adjusted to reflect the risk of default of the counterparty. This is formally referred to as the credit valuation adjustment (CVA). The CVA is the difference between the price of the derivative assuming no default risk and the price when the risk of default is considered.

If the counterparty defaults at time $t_{j-1} \leq \tau \leq t_j$, the bank would lose only if $K(\tau, t_n) > K(t_0, t_n)$ and the exposure would be given by

$$E_j^b = \sum_{i=j}^n \max(K(\tau, t_n) - K(t_0, t_n)) Z(\tau, t_i), \quad (13)$$

where the superscript b is shorthand for bank. E_j^b has the structure of a call option with the strike rate as the underlying, that is, a payer swaption, giving the bank an option to enter a swap as a fixed rate payer. Burgess (2018) shows that the price of a swaption is given by a variation of the Black-76 formula (Black, 1976) such that

$$S(t) = A^{Fixed}(t)K(t, t_n)N(d_1) - K(t_0, t_n)N(d_2), \quad (14)$$

where $A^{Fixed}(t)$ is the swap fixed leg annuity with

$$d_1 = \frac{\ln(K(t, t_n)/K(t_0, t_n)) + (r + \frac{\sigma^2}{2})(t_n - t)}{\sigma\sqrt{t_n - t}}$$

and $d_2 = d_1 - \sigma\sqrt{t_n - t}$.

Altogether, the CVA becomes

$$CVA(t) = (1 - R)S(t)PD(t), \quad (15)$$

where R is the recovery rate, with R being 1 less the loss given default, $R = 1 - LGD$, and $PD(t)$ is the probability of the counterparty's default between time t and the maturity of the swap.

The probability of default can be estimated in various ways. One way is to use market credit default swap (CDS) spreads. Another way would be to use internally defined models, where the bank internally models the probability that the counterparty defaults based on its internal assessments.

Let $\hat{V}$ denote the value of the swap accounting for counterparty default risk, then:

$$\hat{V}(t; t_0, t_n) = V(t; t_0, t_n) - CVA \quad (16)$$

6.3 Carbon valuation adjustment

It is clear from Section 5 that carbon exposure has a material effect on the probability of default of an entity. Thus, it is crucial to incorporate the carbon price risk into the prices of derivatives with counterparties that have carbon exposure. Kenyon et al. (2022) refer to this adjustment as the CO₂-equivalent valuation adjustment (CO₂eVA). Through an extension of Burgard and Kjaer's (2013) xVA approach, Kenyon et al. (2022) use replication to price in the carbon exposure. Burgard and Kjaer (2013) have the following decomposition:

$$\hat{V} = V + U,$$

where V is the Black-Scholes funding and default-free price and U denotes the funding and default adjustments. In the extension, Kenyon et al. (2022) further decompose the adjustment and obtain $\hat{U} = U + G$ where U is still the funding and

default adjustments on the derivative before the carbon exposure consideration and G is the carbon costs inclusive of default and funding effects. Thus, CO₂eVA encapsulates the replication cost or benefit of alleviating CO₂e emissions and the sequestration thereof.

6.3.1 Replication pricing

This paper uses the replication strategy presented in Kenyon et al. (2022). Consider the probability space $(\Omega, \mathcal{F}, \mathbb{Q})$, where Ω is the set of events $\omega \in \Omega$, $\mathcal{F}$ is the sigma-algebra, and $\mathbb{Q}$ is the risk-neutral probability measure that is equivalent to the physical measure $\mathbb{P}$. Due to the possibility of the market being incomplete, $\mathbb{Q}$ could be non-unique. Recall the IRS in Section 6.1, we assume that the derivative references a non-defaultable discount bond P with maturity $\bar{T} \in [T, \infty)$. The price of the derivative accounting for carbon exposure is denoted as $\hat{V}$. In order to reprice the derivative, a replicating portfolio is constructed with instruments with the following processes:

$$dP_{t\bar{T}}/P_{t\bar{T}} = r_t dt + \sigma_{t\bar{T}} dW_t \quad \text{Default-free bond} \quad (17)$$

$$dP_{t\bar{T}}^B/P_{t\bar{T}}^B = r_t^B dt + \sigma_{t\bar{T}}^B dW_t - (1 - R_t^B) dJ_t^B \quad \text{Own bond} \quad (18)$$

$$dP_{t\bar{T}}^C/P_{t\bar{T}}^C = r_t^C dt + \sigma_{t\bar{T}}^C dW_t - (1 - R_t^C) dJ_t^C \quad \text{Counterparty bond} \quad (19)$$

$$dF_t^F/F_t = r_t dt + \sigma_t^F dW_t^F \quad \text{Carbon future contract,} \quad (20)$$

where W_t and W_t^F are correlated standard Wiener processes under $\mathbb{Q}$ with correlation coefficient $\rho_t \in [-1, 1]$. The r_t^B and r_t^C are the funding interest rate process on the bank's own bond and the interest rate process on the counterparty bond, respectively. The $\sigma_{t\bar{T}}$, $\sigma_{t\bar{T}}^B$, and $\sigma_{t\bar{T}}^C$ are the bond volatilities and disappear at bond maturity $\bar{T}$. The jump processes J_t^B and J_t^C , with intensities λ_t^B and λ_t^C , respectively, are independent of each other and of W_t and W_t^F , where $J_0^B = J_0^C = 0$. The bonds default at the first jump of their respective jump processes. In the event of default, the bond is worth the time-dependent but deterministic recovery rate R_t^B , respectively R_t^C .

Kenyon et al. (2022), assume that the carbon future curve has one factor and that there is zero-basis for bond-CDS and bond-repo such that

$$r_t^C - q_t^C = (1 - R_t^C) \lambda_t^C, \quad (21)$$

$$r_t^B - r_t = (1 - R_t^B) \lambda_t^B, \quad (22)$$

where q_t^C is the repo rate on the counterparty bond. Although the price process of the derivative only referenced the default-free bond P , the carbon-linked price process is a function of the default-free bond, the carbon futures contract, on the

default of both the own bond P^B and the counterparty bond P^C . The default of the bonds is tracked by whether either one or both of J_t^B, J_t^C have jumped before the maturity of the derivative, that is in the interval $(0, T]$. Thus, the price process of the carbon-linked derivative is denoted by $\widehat{V}_t(P_{t\bar{T}}, F_t, J_t^B, J_t^C)$.

For ease of notation, the time variables the time variables are omitted, resulting in $\widehat{V}(P, F, J^B, J^C)$ instead of $\widehat{V}_t(P_{t\bar{T}}, F_t, J_t^B, J_t^C)$.

In what follows the replicating portfolio of the interest rate derivative $\widehat{V}$ is priced. First, the change in the value of the derivative is decomposed into its underlying risk factors, giving

$$\begin{aligned} d\widehat{V} = & \frac{\partial \widehat{V}}{\partial t} dt + \frac{\partial \widehat{V}}{\partial P} dP + \frac{\partial \widehat{V}}{\partial F} dF \\ & + \frac{1}{2} \frac{\partial^2 \widehat{V}}{\partial P^2} dP dP + \frac{1}{2} \frac{\partial^2 \widehat{V}}{\partial F^2} dF dF + \frac{\partial^2 \widehat{V}}{\partial P \partial F} dP dF \\ & + \Delta \widehat{V} dJ^B + \Delta \widehat{V} dJ^C, \end{aligned} \quad (23)$$

where

$$\begin{aligned} \Delta_B \widehat{V} &= g^B - \widehat{V}, \\ \Delta_C \widehat{V} &= g^C - \widehat{V} \end{aligned} \quad (24)$$

are the spreads due to the closeout costs g^B and g^C at default. To replicate $\widehat{V}$, a replicating portfolio Π is given by:

$$\Pi = \delta P + \alpha^B P^B + \alpha^C P^C + \alpha^F F + \beta + \beta^C + \beta^X + \beta^F, \quad (25)$$

where $\delta, \alpha^B, \alpha^C$, and α^F reflect the number of units of each of the instruments. The betas denote the cash accounts associated with the bank bond P , the repo rate on P^B , the derivative collateral, X and the margin on the futures contract. Before the hedge is rebalanced, the changes on the cash accounts are given as:

$$\begin{aligned} d\beta &= -\delta r P dt, & d\beta^C &= -\alpha^C q^C P^C dt, \\ d\beta_X &= -r^X X dt, & d\beta^F &= -\alpha^F 0 F dt. \end{aligned}$$

Since the margin on futures contracts does not earn interest, there is no differential cashflow associated with the futures cash account, that is, $d\beta^F=0$. Assuming that the changes in the value of the replicating portfolio Π emanate from the changes in the price processes of the instruments in the portfolio, the sum of the differentials and of the replicating portfolio and the interest rate derivative instrument is given

by

$$\begin{aligned}
d\hat{V} + d\Pi = & \frac{1}{2}\sigma^2 P^2 \frac{\partial^2 \hat{V}}{\partial P^2} dt + \frac{\partial \hat{V}}{\partial P} dP + \frac{\partial \hat{V}}{\partial t} dt \\
& + \frac{1}{2}\sigma^{F^2} F^2 \frac{\partial^2 \hat{V}}{\partial F^2} dt + \frac{\partial \hat{V}}{\partial F} dF + \rho\sigma^F \frac{\partial^2 \hat{V}}{\partial P \partial F} dP dF \\
& + \Delta \hat{V} dJ^B + \Delta \hat{V} dJ^C \\
& + \delta dP + \alpha^B dP^B + \alpha^C dP^C + \alpha^F dF \\
& - \delta r P dt - \alpha^C q^C P^C dt - r^X X dt.
\end{aligned} \tag{26}$$

To find the hedge ratios for the hedge, each of the terms dW , dW^F and dJ^C are set to zero. However, the hedge for the dJ^B term is not perfect, so it is not set to zero.

1. To hedge the carbon futures price risk, (dW^F -term) by setting $\sigma^F F \partial_F \hat{V} + \alpha^F \sigma^F F = 0$. Thus, $\alpha^F = -\partial_F \hat{V}$.
2. Hedging the counterparty default risk (dJ^C -term), $-\alpha^C(1 - R^C)P^C + \Delta_C \hat{V} = 0$. It follows,

$$\alpha^C = \frac{g^C - \hat{V}}{(1 - R^C)P^C}. \tag{27}$$

3. To hedge against the risk arising from P , P^B , and P^C , the dW -term is set to zero:

$$\alpha^B P^B \sigma^B + \alpha^C P^C \sigma^C + \delta \sigma P + \sigma P \partial_P \hat{V} = 0, \tag{28}$$

substituting the value of α^C from (27) we get

$$\delta = -\partial_P \hat{V} - \alpha^B \frac{P^B}{\sigma P} - \frac{\sigma^C (g^C - \hat{V})}{\sigma P (1 - R^C)}. \tag{29}$$

4. For the dJ^C -term, a partial hedge is constructed such that there is exposure to a cost (or benefit) of $\epsilon \neq 0$, $-\alpha^B(1 - R^B)P^B + \Delta_B \hat{V} = \epsilon$ and choose that

$$\epsilon = (1 - R^B)(\hat{V} - X) + g^B - \hat{V}, \tag{30}$$

where α^B is found by requiring that the net cash in the hedge position is funded by issuing or buying back one's own bonds P^B used in the replicating portfolio II. This requires $(\hat{V} - X) + \alpha^B P^B = 0$. Following the usual semi-replication strategy, cashflows up to, but not including the bank's own default are replicated. Substituting the value of α^B into (29), the amount invested in the default-free discount bond is given by

$$\delta = -\partial_P \hat{V} + \frac{1}{\sigma P} \left[\sigma^B (\hat{V} - X) - \frac{\sigma^C \Delta_C \hat{V}}{(1 - R^C)} \right]. \tag{31}$$

5. For no arbitrage, $d\hat{V} + d\Pi = 0$ for all $t \in [0, T]$, therefore, the overall dt term must be equal to zero:

$$\begin{aligned}
& \partial_t \hat{V} + \frac{1}{2} \sigma^2 P^2 \partial_{PP} \hat{V} + \frac{1}{2} \sigma^{F2} F^2 \partial_{FF} \hat{V} + \rho \sigma \sigma^F P F \partial_{PF} \hat{V} \\
& - \alpha^C q^C P^C - r^X X + \alpha^B P^B r^B + \alpha^C P^C r^C + r P \partial_P \hat{V} + r F \partial_F \hat{V} \\
& = \partial_t \hat{V} - (r^B + \lambda^C) \hat{V} + (r^B - r^X) X + \lambda^C g^C \\
& + \underbrace{\frac{1}{2} \sigma^2 P^2 \partial_{PP} \hat{V} + \frac{1}{2} \sigma^{F2} F^2 \partial_{FF} \hat{V} + \rho \sigma \sigma^F P F \partial_{PF} \hat{V} + r P \partial_P \hat{V} + r F \partial_F \hat{V}}_{\mathcal{A}_t \hat{V}} \\
& = 0
\end{aligned} \tag{32}$$

We then obtain the PDE:

$$\begin{cases} \partial_t \hat{V} + \mathcal{A}_t \hat{V} = (r^B + \lambda^C) \hat{V} + (r^B - r^X) X + \lambda^C g^C, \\ \hat{V}(T, P, F) = \hat{V}(P, F). \end{cases} \tag{33}$$

Under the assumption of no defaults by time $t \in (0, T]$, $\hat{V}$ solves this PDE. The market-traded derivative, V , that does not include carbon price risk solves the PDE:

$$\begin{cases} \partial_t V + \beta_t V - rV = 0, \\ V(T, P) = V(P), \end{cases} \tag{34}$$

Where $V(P)$ is the payoff of the derivative at maturity T . The β_t is a one-dimensional version of $\mathcal{A}_t$ because it does not include the terms that depend on F .

To put everything together, the decomposition $\hat{V} = V + \hat{U}$ is used, where $\hat{U}$ is the total value adjustment that emanates from the carbon price risk, that is, $\hat{U}$ is added to the market price of the derivative price V to get the carbon-risk adjusted price $\hat{V}$.

PDE (33) can be written as

$$\begin{cases} \partial_t \hat{U} + \mathcal{A}_t \hat{U} - (r^B + \lambda^C) \hat{U} = (r^B - r) V - (r^B - r^X) X + \lambda^C (V - g^C), \\ \hat{U}(T, P, F) = \hat{V}(P, F) - V(P). \end{cases} \tag{35}$$

To isolate the valuation adjustment arising from the carbon price risk, $\hat{U}$ is decomposed further into $\hat{U} = U + G$. This decomposition allows us to obtain a PDE that governs the dynamics of the CO₂e-valuation adjustment. We can then write PDE (35) as

$$\begin{cases} \partial_t U + \mathcal{B}_t U - (r^B + \lambda^C)U + \partial_t G + \mathcal{A}_t G - (r^B + \lambda^C)G \\ \quad = (r^B - r)V - (r^B - r^X)X + \lambda^C(V - g^C), \\ U(T, P) + G(T, P, F) = G(P, F). \end{cases} \quad (36)$$

The function U is chosen such that it satisfies the PDE:

$$\begin{cases} \partial_t U + \mathcal{B}_t U - (r^B + \lambda^C)U - (r^B - r)V + (r^B - r^X)X - \lambda^C(V - g^C) = 0 \\ U(T, P) = 0. \end{cases} \quad (37)$$

The $U(T, P)$ is the value adjustment to the derivative V , excluding carbon price risk, that arises from counterparty risk and funding constraints emanating from the replicating instruments P, P^B , and P^C .

To get the value adjustment that coincides with the carbon price risk, we use the decomposition stipulated in (36) and the constraint (37), and we require that G satisfies:

$$\begin{cases} \partial_t G + \mathcal{A}_t G - (r^B + \lambda^C)G = 0, \\ G(T, P, F) = G(P, F). \end{cases} \quad (38)$$

Therefore, we more formally relate the CO₂e-valuation adjustment with G by

$$\text{CO}_2\text{eVA}(t, P, F) := G(t, P, F). \quad (39)$$

The CO₂e-valuation adjustment can also be expressed as

$$\text{CO}_2\text{eVA}(t, T) = \int_t^T \mathbb{E}[D(u)x(u)g(V(u), F(u))|\mathcal{F}_t]du, \quad (40)$$

where $x(u)$ is a scaling factor to account for counterparty emissions. The factor increases linearly and is assumed to be zero today. $D(t)$ is the discount factor with rate $r^B + \lambda^C$. The $g(\cdot)$ represents the charge function and depends on the counterparty net participation charge.

Collectively, to express the CO₂eVA(t, T) in value adjustment format we have:

$$\widehat{V}(t, P, F) = V(t) + x\text{VA}(t) + \text{CO}_2\text{eVA}(t). \quad (41)$$

6.4 Eskom IRS Example

Using the setup in Kenyon et al. (2022), we consider the CO₂e charge associated with an IRS traded between a bank and a utility company Eskom. Given that Eskom emits 207 million tons of carbon per year, the carbon price risk that comes with these emissions would have a material impact on the value of the IRS.

1. The participation charge function $g(\cdot)$ depends on the carbon price $F_i(t)$ in each of the NGFS scenarios.

2. All five scenarios are given equal weights.
3. The net participation charge of the IRS is a function of the net contribution to the loans on the balance sheet of the counterparty Eskom. This is because of the assumption that loans are floating rate and swapped to fixed using the IRS.
4. The time $x(t)$ at which the charge is introduced is zero today and increases linearly to one until 2050 and remains constant thereafter.
5. The carbon prices are given in NGFS in USD re-based to 2010. They are then converted to 2023 using US inflation, then EU inflation thereafter to current prices.

Data description

- 03 January 2023
- IRS at-the-money of maturity 5 years to 20 years.
- Notional of R100 million.
- Interest rates are taken as Jibar 3m single curve setup.
- Interest rates are taken as coterminal ATM swaption normal volatilities
- Eskom's loans were R453.8 billion in 2022.
- The Eskom CDS spreads are used for default probability

Table 6: CO₂eVA for IRS for Eskom.

Maturity	CVA(rec)	FVA	CVA(payer)	CO ₂ eVA(epe)(bps)	$\frac{\text{CO}_2\text{eVA}(\text{epe})}{\text{CVA}(\text{rec})}$	$\frac{\text{CO}_2\text{eVA}(\text{ev})}{\text{FVA}}$	$\frac{\text{CO}_2\text{eVA}(\text{ene})}{\text{CVA}(\text{payer})}$
5	18	2	24	26	1.07	5.57	-0.98
7	33	5	54	80	1.47	8.02	-1.30
10	60	9	103	202	1.97	10.86	-1.81
15	108	11	165	451	2.73	13.89	-2.73
20	147	12	206	689	3.35	14.69	-3.54

Table 6 shows that as the maturity increases, the CO₂eVA becomes higher multiples of the CVA and FVA costs. This is because as the maturity increases, the carbon prices and the participation factor $x(t)$ also increase.

7 Conclusions

This research focuses on quantifying the impact produced by Eskom's carbon footprint on its probability of default. The analysis relies on an asset-liability model for Eskom's firm valuation inspired by Merton's credit risk pricing approach. We produce a number of simulations for the evolution of Eskom's running default probability based on South Africa's IRP 2019 and the NGFS carbon price scenarios. Since Eskom's asset base depends on the revenues resulting from the sale of electricity, we propose a stochastic electricity sell price process and derive the running minimum electricity price to cover all production costs in each time period. The report concludes with calculating the CO₂e valuation adjustment to a swap contract between Eskom and a bank.

Bibliography

- Battiston, S., Dafermos, Y., Monasterolo, I., 2021. Climate risks and financial stability.
- Black, F., 1976. The pricing of commodity contracts. *Journal of financial economics* 3, 167–179.
- Blaufelder, C., Levy, C., Mannion, P., Pinner, D., 2021. A blueprint for scaling voluntary carbon markets to meet the climate challenge. *McKinsey* 5.
- Boffo, R., Patalano, R., 2020. ESG investing: practices, progress and challenges. Éditions OCDE, Paris .
- Bolton, P., Kacperczyk, M., 2021. Do investors care about carbon risk? *Journal of financial economics* 142, 517–549.
- Burgard, C., Kjaer, M., 2013. Funding costs, funding strategies. *Risk* , 82–87.
- Burgess, N., 2018. Interest rate swaptions: a review and derivation of swaption pricing formulae. *Journal of Economics and Financial Analysis* 2, 87–103.
- Climate Action Tracker, 2023. Climate action tracker data portal. Retrieved from: <https://climateactiontracker.org/countries/south-africa/net-zero-targets/>.
- Climate Watch, 2020. Greenhouse gas emissions. Retrieved from:<http://surl.li/izxqu>.
- DGB Group, 2023. Understanding carbon credit futures: an overview of the carbon market. Retrieved from:<https://www.green.earth/blog/understanding-carbon-credit-futures-an-overview-of-the-carbon-market>.
- ESKOM, 2022. Annual financial statements. Retrieved at: https://www.eskom.co.za/wp-content/uploads/2022/12/2022_annual_financial_statements.pdf.

- ESKOM Generation Division, 2021. Generation plant mix Retrieved at: <https://www.eskom.co.za/wp-content/uploads/2022/03/GX-0001-Generation-Plant-Mix-Rev-25.docx.pdf>.
- European Commission, 2019. Annual Digest 2021 – SA-EU trade under the SADC EU EPA.
- European Commission, 2023. Questions and answers: carbon border adjustment mechanism (CBAM). Retrieved from: <https://taxation-customs.ec.europa.eu/system/files/2023-06/20230602%20Q%26A%20CBAM.pdf>.
- Gandhi, V.P., Cuervo, J., 1998. Carbon taxes: Their macroeconomic effects and prospects for global adoption: A survey of the literature .
- Hayhoe, K., Wuebbles, D.J., Easterling, D.R., Fahey, D.W., Doherty, S., Kossin, J.P., Sweet, W.V., Vose, R.S., Wehner, M.F., 2018. Our changing climate. Impacts, risks, and adaptation in the United States: The Fourth National Climate Assessment, Volume II, Chapter 2. Technical Report. URL: <https://doi.org/10.7930%2Fnca4.2018.ch2>, doi:10.7930/nca4.2018.ch2.
- Inderst, G., Stewart, F., 2018. Incorporating environmental, social and governance (ESG) factors into fixed income investment. World Bank Group publication, April .
- IRP, 2019. South Africa integrated resource plan. Retrieved from: <https://www.energy.gov.za/irp/2019/IRP-2019.pdf>.
- Keen, M., Parry, I., Roaf, J., 2022. Border carbon adjustments: rationale, design and impact. *Fiscal Studies* 43, 209–234.
- Kenyon, C., Berrahoui, M., Macrina, A., 2021. The carbon equivalence principle: minimizing the cost to carbon net zero. Available at SSRN 3979608 .
- Kenyon, C., Macrina, A., Berrahoui, M., 2022. CO2eVA: pricing the transition of carbon externalities. Available at SSRN 4136710 .
- Kenyon, C., Macrina, A., Berrahoui, M., 2023. The carbon equivalence principle: methods for project finance. *Risk, Cutting Edge: Climate Finance*, Risk.net .
- Mallucci, E., 2022. Natural disasters, climate change, and sovereign risk. *Journal of International Economics* 139, 103672.
- Mashishi, N., 2012. South Africa the 12th biggest source of greenhouse gases? Yes, but that's not the only measure that matters. Retrieved from: <https://shrturl.app/Tm2CPT>.

- Matsheta, R.M., Sefoka, I.M., 2023. Load-shedding in South Africa: an immediate threat to the right to education, "Section 29 Inquiry". Journal of Educational and Social Research 13.
- Merton, R.C., 1974. On the pricing of corporate debt: The risk structure of interest rates. The Journal of Finance 29, 449–470.
- National Energy Regulator of South Africa, 2023. The Decision and Reasons for Decision .
- NGFS, 2022. NGFS scenarios for central banks and supervisors, September 2022. Retrieved from: <https://www.ngfs.net>.
- Oprean-Stan, C., Oncioiu, I., Iuga, I.C., Stan, S., 2020. Impact of sustainability reporting and inadequate management of esg factors on corporate performance and sustainable growth. Sustainability 12, 8536.
- Parry, I., 2019. What is carbon taxation? Finance & Development, 8.
- Portal, N., . NGFS Phase 3 Scenario Explorer. Retrieved from: <https://data.ene.iiasa.ac.at/ngfs/#/workspaces/24>.
- SA Department of Environmental Affairs, 2020. South Africa's low-emission development strategy 2050. Retrieved at: https://www.dffe.gov.za/sites/default/files/docs/2020lowemission_developmentstrategy.pdf.
- The World Bank, 2023. What is carbon pricing? Retrieved from: <https://www.worldbank.org/en/programs/pricing-carbon>.
- USAID, 2015. Greenhouse gas emissions in Southern Africa. Climate Links. Retrieved at: https://pdf.usaid.gov/pdf_docs/pa00msrh.pdf.
- Wang, Y., 2009. Structural credit risk modeling: Merton and beyond. Risk Management 16, 30–33.