Tail Risk and Uncertainty in Financial Markets

by

Robert J. Millar

A thesis submitted to
The University of Birmingham
for the degree of
Doctor of Philosophy

School of Mathematics
University of Birmingham
September 2024

"Uncertainty is an uncomfortable position.

But certainty is an absurd one."

– *Voltaire*

Abstract

This thesis explores the intricacies of managing and modelling tail risk and uncertainty in financial markets. Tail risks arise from infrequent but potentially significant events, while uncertainty refers to the unpredictable aspects of market dynamics that cannot be accounted for by standard probabilistic models. Traditional models often struggle to account for these elements, leading to inadequate investment strategies which underestimate risk. This thesis proposes a new investment framework centred around three objectives: (1) to develop forecasting methods that provide more accurate and robust predictions of asset prices and volatility, accounting for the inherent uncertainty in financial markets (2) to devise new methodologies which better predict the probability and impact of tail events (3) to create portfolio allocation algorithms which eliminate unrealistic assumptions and better reflect the complex dynamics of modern financial markets.

In addressing the first objective, this thesis critiques existing forecasting models and introduces a Bayesian approach to ARMA-GARCH modelling. This new approach incorporates prior knowledge and directly accounts for the uncertainty in financial data, offering a more robust prediction framework.

Regarding the second objective, this thesis introduces existing quantitative tools to measure financial risk and then proposes a new algorithm called the *Multicanonical Sequential Monte Carlo Sampler* (MSMCS), which efficiently reconstructs prob-

ability distributions to capture tail risk.

For the final objective, this thesis proposes a series of Bayesian Optimisation algorithms that address optimal portfolio allocation. These algorithms are tailored to reduce the computational intensity often associated with such tasks and to take advantage of specific characteristics of portfolio optimisation problems.

This thesis culminates in the combined application of the Bayesian ARMA-GARCH models to forecast asset returns, MSMCS to assess tail risk, and Bayesian Optimisation to find an optimal portfolio allocation. The combined framework is applied to historic market data and shown to outperform various existing strategies and market indices.

This work contributes to financial mathematics by challenging conventional approaches and introducing new Bayesian-based models that more accurately reflect the complexity and inherent uncertainties of financial markets. It provides a foundation for further research and practical applications in financial forecasting models, risk assessment, and portfolio management.

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Jinglai Li, for all of his support and advice throughout my PhD. I have been extremely lucky to have a supervisor so generous with his time, guidance and advice, supporting my research and academic progression.

I am grateful to have completed my undergraduate studies and postgraduate research at the University of Birmingham. In particular, I want to thank my undergraduate tutor, Dr Simon Goodwin, and collaborators, Dr Hui Li and Dr Jia Shao.

Last but not least, I thank my family and friends for their support and encouragement throughout my studies.

Contents

Chapter 1

Financial Markets

1.1 Overview of financial markets

In the realm of financial markets, it is essential to distinguish between real assets which contribute directly to an economy's productive capacity (like land, buildings, machinery, human capital, knowledge, etc.) and financial assets (like stocks and bonds) which serve as a means for companies to raise capital for the acquisition and development of real assets. Financial assets also represent claims on the income and value of real assets. For example, while one may not directly own a significant industrial asset like a mining operation (a real asset), by purchasing shares (a financial asset) in a corporation engaged in mining activities, one can benefit from the revenue generated by the company's production activities from such real assets.

1.1.1 Sell-side dynamics

On the sell-side of financial markets, Investment Banks (IBs) assist companies in formulating a balanced capital structure to finance their operations and growth. This includes bond issuance to fund investments expected to yield returns surpassing the debt servicing costs (i.e. interest and principal payments). The bonds are structured into tranches whereby, in periods of financial difficulty, a company pays

debt holders in the order of tranch seniority (senior, junior, and then mezzanine). This approach allows companies to optimise financing costs and risk management—where the issuance prices and rates are set per tranche to reflect the company's credit risk, ensuring that investors receive appropriate compensation for the level of risk undertaken.

Investors in equity gain partial ownership in the companies; deriving returns from dividends and changes in stock value. Regarding a company's overall capital structure, equity investors are the first risk bearers in periods of financial distress. Debt holders face default only after all possible losses are absorbed by equity holders. Whilst equity may provide the highest potential returns, it also has the greatest risk. Finally, beyond bonds and stocks, derivatives like options, futures, and swaps can be used for risk hedging and speculative purposes. For example, a company might use futures to stabilise raw material costs against fluctuating commodity prices, or swaps to guard against interest rate changes, enabling more predictable financial management.

1.1.2 Buy-side dynamics

On the buy-side of financial markets, investors acquire and manage financial assets, aiming to balance investment risk and return based on their specific investment goals. Mutual funds pool resources from many investors to build diversified portfolios, targeting various return objectives and risk profiles. Pension funds invest employees' retirement savings, prioritising long-term, stable returns in order to meet future pension liabilities. Insurance companies invest premium income to ensure they can cover future claims, often adopting a more conservative strategy with a mix of low-risk bonds and some other securities for higher returns. With differing investment strategies, all three contribute to the dynamics and liquidity of financial markets.

As an example, when a company initiates a new project, the future cash flows from this venture remain uncertain. Suppose the company gathers capital for its new project by issuing both equity and bonds. Some investors might purchase the equity, embracing the associated risk and potential for higher returns, while more cautious investors might opt for bonds, which promise more predictable returns. As such, the project's risk is distributed amongst the investors, becoming a financial risk, which is then priced and managed by the investors as part of their overall strategies.

The variety of financial instruments enables investors with higher risk tolerance to shoulder more of that risk in pursuit of higher returns, whereas those with a lower risk appetite can opt for more secure assets. This facilitates more tailored investment strategies for investors, catering to their unique risk profiles and return expectations, and it enables corporations to access diverse funding sources at the best rates, further bolstering the real assets in the economy.

1.1.3 Financial markets uncertainty

Financial markets serve a dual role: to enable companies to access various funding sources for real asset growth, and to allow investors to achieve their financial goals. The collective actions of the buy and sell side ensure market prices accurately represent risk, and as investors manage this risk as part of their broader investment strategy, financial markets can be viewed as self-regulating.

This idealised view of financial markets is the foundation of many mathematical models. In reality, financial markets are fraught with uncertainty. Investors often make irrational decisions based on emotions and biases. Market prices can be distorted by speculation, misinformation, and herd mentality rather than reflecting information accurately. Financial markets do not perfectly self-regulate.

Moreover, investor forecasting activity has historically focused on predicting frequent but relatively unimportant events, e.g. as shown by Bond and Dow (2021). Consequently, investors often only bet on or against frequent events, limiting their potential gains and resulting in a failure to foresee rare events, which tend to have the largest financial implications. The historical record is littered with such events, also called *tail events*; a prime example is the Global Financial Crisis (GFC) of 2007-09, which resulted in asset valuation write-downs totalling c. $4.1 trillion across the USA, Europe, and Japan (IMF, 2009).

This thesis focuses on the buy side and optimal investment strategies. The following section gives an overview of the evolution of such investment strategies, highlighting how irrational behaviours, market inefficiencies, and unexpected events challenge the idealised perception of financial markets.

1.2 The evolution of investment decisions

Financial modelling and investment decision-making have evolved significantly since the early 20th century, challenging the idealised view of financial markets. In the 1930s, Economists like Fisher laid the groundwork for stock market analysis using complex mathematical models. Whilst useful, these early models are overly simplistic in addressing real market complexities and investor behaviours.

The 1950s saw the development of the Modern Portfolio Theory (MPT) by Markowitz (1952), introducing the concept of diversification and the first mathematical framework for optimal investing. MPT provides a useful theoretical foundation for optimal investing. However, the key assumption of asset returns being uncorrelated to each other and investor behaviour being rationale is unrealistic, particularly during market crashes and volatile conditions (Curtis, 2004).

The advent of computer-based models in the 1950s and 1960s further revolu-

tionised financial analysis, enabling the processing of large datasets and the development of technical analysis methods like trend line analysis and indicators such as the Relative Strength Index (Bodie et al., 2002).

In the 1970s, Fama's Efficient Market Hypothesis (Fama, 1970) posited that markets and asset prices reflect all available public and private information, negating the possibility of consistently outperforming the market. This hypothesis has been largely rebuked; for example, many investors have consistently outperformed market averages, indicating the existence of inefficiencies in market pricing (Ball, 1996). Later, Kahneman and Tversky (1979) studied and formalised such inefficiencies through their work on Behaviour Finance. Their work challenges the rational investor assumption underlying most traditional financial models, highlighting investor's tendency to favour long shots, avoid near-certain gains, and take large risks to win back losses (Curtis, 2004).

The late 20th and early 21st centuries marked a shift towards more sophisticated financial models to accommodate the growing complexity of markets and advanced investment strategies (Bodie et al., 2002). The 'quantitative revolution' continues to this day, with over 80% of trading on the NYSE becoming automated through mathematical algorithms by the late 2010s (CNBC, 2019).

1.2.1 Limitations of existing models

Uncertainity

More than a century ago, Knight (1921) emphasised the importance of distinguishing between quantifiable risk and unquantifiable uncertainties in financial markets. Later, Keynes (1936) related this directly to probability theory, explicitly distinguishing *risk* as quantifiable uncertainties that can be measured through probability, from *true uncertainty* representing those outcomes that are unknown and not

amenable to probabilistic prediction. Whilst most investors today consider risk in their investment decisions, few consider uncertainty.

The importance of this distinction is central to the recent publication by Kay and King (2020), which challenged the standard mathematical models used in economics and finance, arguing such models are too reliant on the ideas of rationality and predictable outcomes, with no regard for uncertainty. The work calls for mathematicians and financial modellers to recognise the limitations of purely probabilistic approaches and emphasise the need for more robust and flexible mathematical models that perform well under a wide range of scenarios, adapting to unforeseen changes and conditions. It also calls for the incorporation of prior or expert knowledge.

Keynes (1921) and Knight (1921) also highlighted the challenges in formulating uncertainty in terms of precise probabilities, emphasising the paradox that the future's unpredictability is both a source of problems and solutions in knowledge acquisition. In his exploration of econometric models, Hansen (2014) challenged the reliance solely on historical data in predicting future events, suggesting models should incorporate uncertainty as an integral component rather than relying solely on deterministic or probabilistic forecasts. Furthermore, Hansen and Sargent (2001), Hansen and Sargent (2007) and Hansen (2014) explore the importance of robust control and model uncertainty in economic models, arguing for new models that can accommodate the complexities and ambiguities inherent in financial systems.

The limitations of traditional econometric approaches, particularly in handling uncertainty, are further emphasised by the works of Ai and Chen (2003), Chamberlain (1987), and Newey (1990). Chen and Epstein (2002) discuss ambiguity and risk in asset returns, indicating a gap in standard models in addressing uncertainties in continuous time. This body of work underscores the critical need for financial models beyond simple deterministic and probabilistic frameworks, which more accurately

reflect financial markets' inherent uncertainties and complexities.

Tail risk

Incorporating tail risk into financial modelling and portfolio optimisation is a critical aspect of modern financial analysis, as underscored by the growing body of literature on the subject. Traditional models' inability to capture the impact of rare, unforeseeable, and significant events, collectively referred to as "Black Swan" events, is studied in Taleb (2005). Gao and Song (2018) further emphasise the ubiquitous nature of tail risk across financial markets, indicating its pervasive impact on market dynamics and investor behaviour.

The importance of tail risk is highlighted in the works of Albagli et al. (2015), who demonstrate how heterogeneous information[1] among market participants influences asset pricing, particularly in the context of extreme events. Underscoring the need for financial models to account for these risks, Kelly and Jiang (2014) discuss how tail risk significantly influences asset prices. This need for inclusion is echoed in works focusing on non-Normal distributions and extreme events in economic models, for example, in studies by Breon-Drish (2015), Straub and Ulbricht (2021) and Chabakauri et al. (2021). By acknowledging and quantifying tail risks, investors can better prepare for unforeseen market disruptions, allocate resources more effectively, and better understand the potential risks and returns in financial markets.

The combination of modelling uncertainty, a key component in preparing for tail risk, and the role of "Black Swan" events are explored in the works of Orlik and Veldkamp (2014), and Cogley and Sargent (2005). Integrating uncertainty and tail risk into financial modelling and investment strategies is paramount in navigating the complexities and uncertainties inherent in today's financial markets.

[1]In financial markets, not all participants have access to the same information or interpret information in the same way. This disparity is referred to as "heterogeneous information". The impact of this becomes particularly pronounced during extreme events (Albagli et al., 2015).

1.2.2 The need for a new approach

Given the evidence against the notion of inherently efficient and self-regulating markets and the importance of incorporating uncertainty and tail risk into financial models, this thesis proposes a new approach to financial market analysis and investment decision-making. To address the current limitations, this thesis proposes a comprehensive re-evaluation of the investment decision process—across returns forecasting, risk management and portfolio optimisation—centred around three objectives:

1. **Asset Returns Forecasting.** Develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

2. **Risk Management.** Develop new methodologies which better predict the probability and impact of tail events.

3. **Optimal Portfolio Allocation.** Develop portfolio allocation algorithms which better reflect the complex dynamics of modern financial markets, removing unrealistic assumptions.

Bayesian modelling may address these challenges. Unlike traditional methods, Bayesian models provide a probabilistic understanding of model parameters, offering deeper insights into uncertainty. Bayesian models can incorporate prior knowledge and expert judgements, and update predictions with new information, making them adaptable and adept at handling tail risk.

This thesis is structured around these three core objectives. We will examine existing approaches to each objective and then present our proposed methods and algorithms to achieve the aims. We demonstrate the proposals using both rigorous mathematical theory and numerical examples.

1.3 Key data

For the sake of simplicity and to ease evaluation, we introduce a central task: determining the optimal allocation to a selection of nine assets. This objective is achieved through improved returns forecasting, risk management and optimisation algorithms. The nine assets are carefully selected to encompass a cross-section of sectors, market capitalisations, and regions (United States of America (US), United Kingdom (UK), and People's Republic of China (CN))—summarised in Table 1.1.

The equity index with the largest total market capitalisation within each country is chosen, as it best represents the stock market performance for that country. Within each index, the company with the largest and smallest market capitalisation are selected unless the sector has already been covered, in which case the next largest (or smallest) company is chosen.

Country	**Asset**	**Ticker**	**Sector**	**Market cap**
US	S&P 500	SP500	Index	-
	Apple Inc	AAPL	Technology	$2.9tn
	Alaska Air Group Inc	ALK	Aviation	$6.6bn
UK	FTSE 100	FTSE100	Index	-
	Shell Plc	SHEL	Energy	$89.7bn
	Taylor Wimpey Plc	TW	Construction	$8.6bn
CN	SSE Composite Index	SSE	Index	-
	Ping An Insurance Ltd	PING	Insurance	$139.3bn
	Shenzhen Salubris Pharmaceuticals Co Ltd	SHEN	Pharmaceuticals	$29.6bn

Table 1.1: Key information on core dataset. Market capitalisation data, as at End-2021.

Historical asset price data can give gross and net returns, which show the actual percentage change in stock prices. Additionally, one can obtain logarithmic returns, which have several advantages, including stabilising return series variance, statistical properties like symmetry and time additivity, and the accommodation of compounding effects. Formally, the returns are defined as follows:

Definition 1.3.1 *(Tsay, 2005) Let P_t be the value or price of an asset at time t. Then the 'simple gross return', denoted by $1 + R_t$, for an asset held over the period from date $t-1$ to date t is:*

$$1 + R_t = \frac{P_t}{P_{t-1}}. \tag{1.3.2}$$

The corresponding 'simple net return', denoted by R_t, is:

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}}. \tag{1.3.3}$$

Definition 1.3.4 *(Tsay, 2005) The natural logarithm of the simple gross return is called the 'log return', denoted by r_t, calculated as:*

$$r_t = ln(1 + R_t) = ln\left(\frac{P_t}{P_{t-1}}\right) = p_t - p_{t-1}, \tag{1.3.5}$$

where $p_t = ln(P_t)$ is the natural logarithm of the asset price at time t.

For the purpose of this work, we focus on log returns—given the aforementioned advantages. For the core data set, we provide a visualisation of the historical prices (on linear and log scales) and weekly returns[2].

[2]A forward-filling method was employed, ensuring continuity in the dataset to address any missing values and maintain data integrity.

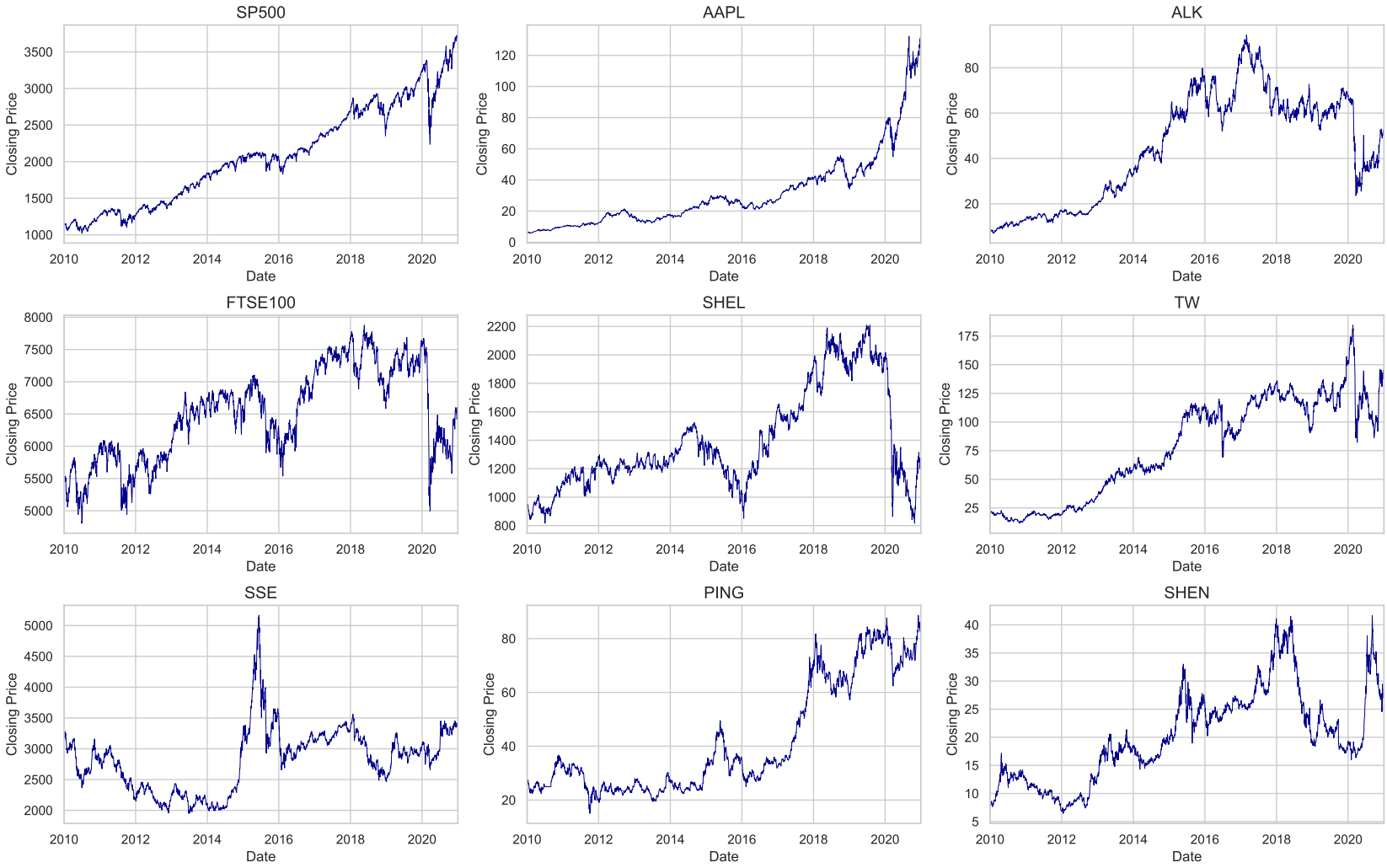


Figure 1.1: Historic prices (linear scale) from January 1, 2010 to December 30, 2021.

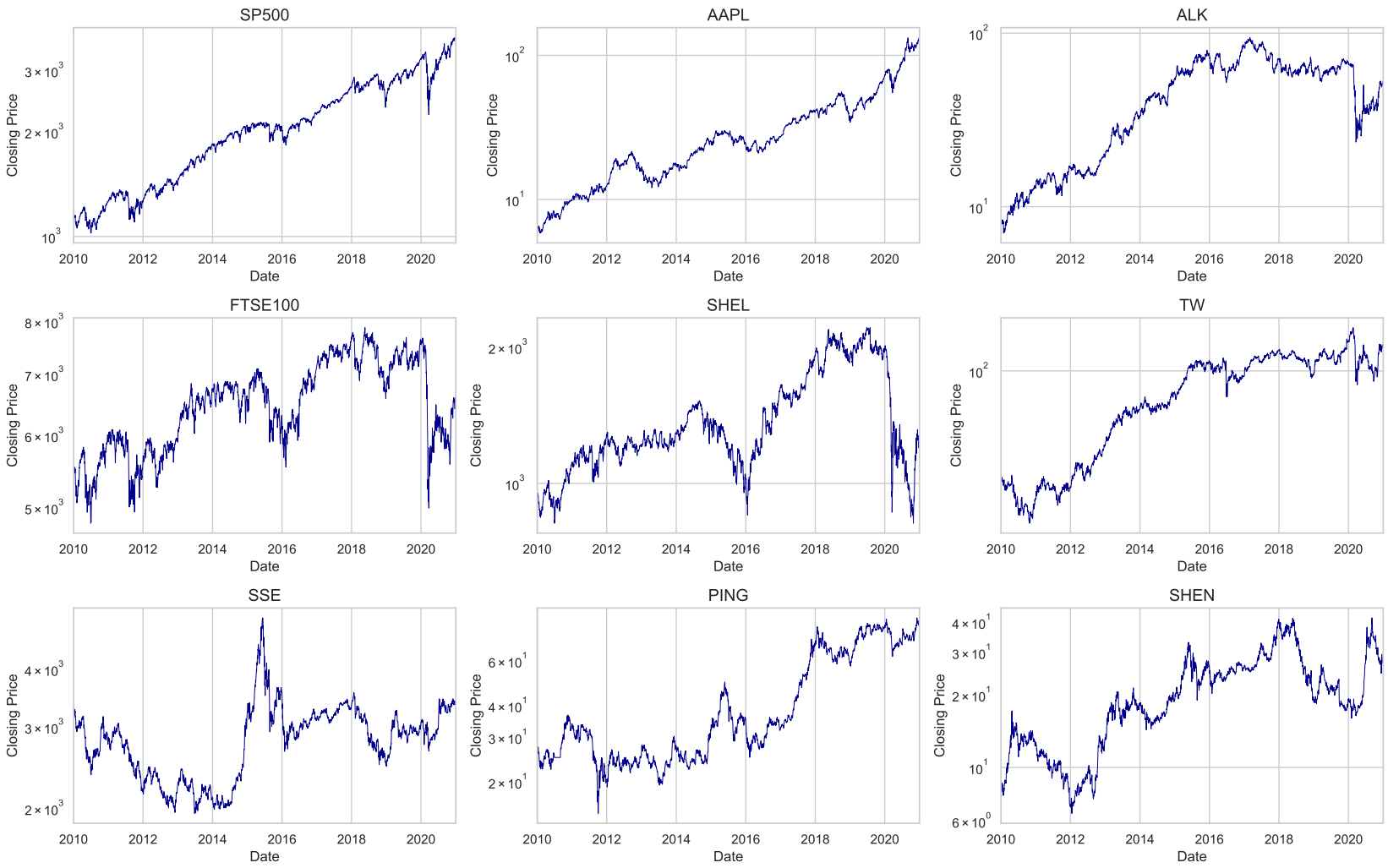


Figure 1.2: Historic prices (log scale) from January 1, 2010 to December 30, 2021.

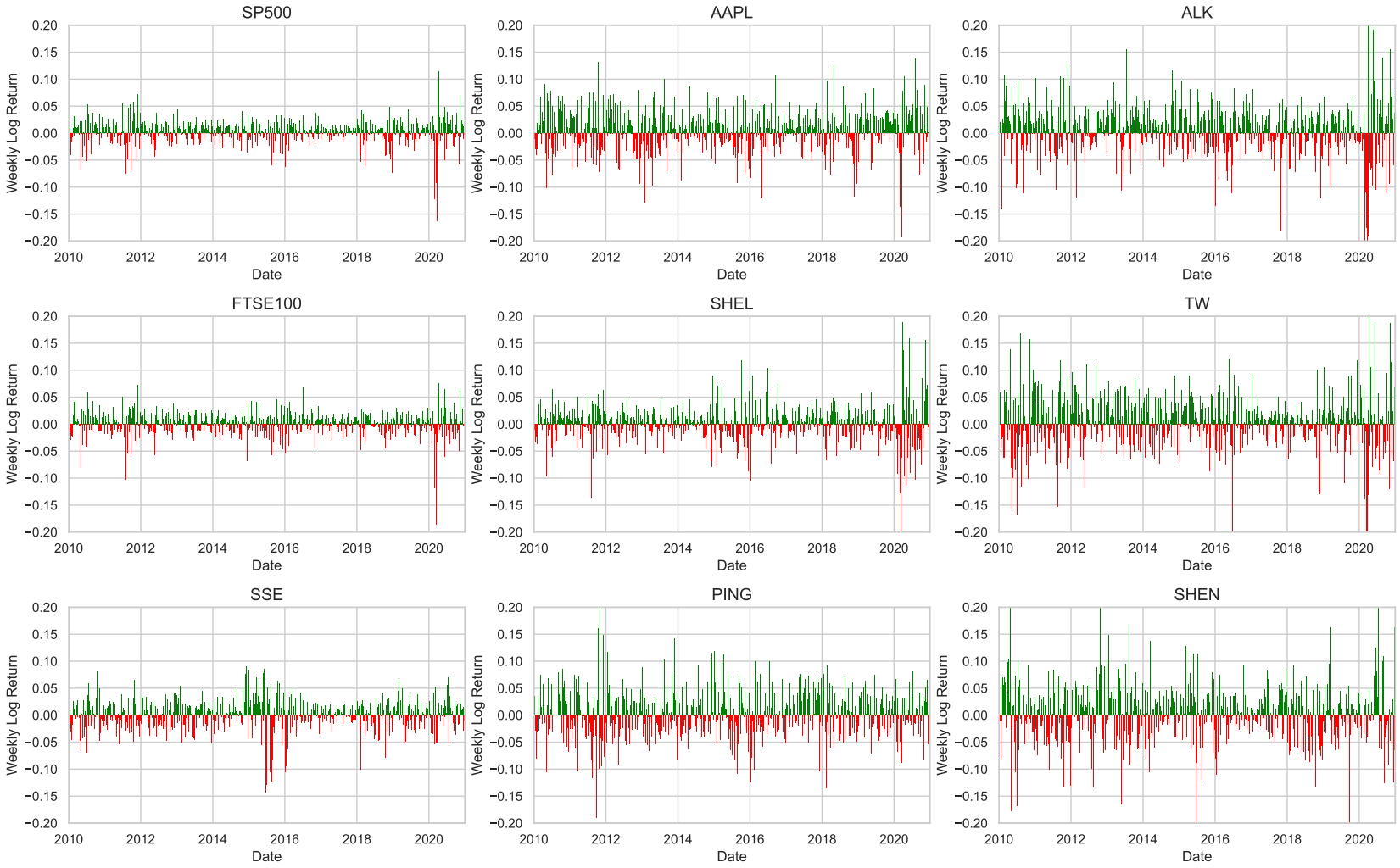


Figure 1.3: Historic weekly log returns from January 1, 2010 to December 30, 2021.

1.3.1 Preliminary observations

The U.S. stock market, represented by the S&P 500 index (SP500), Apple Inc. (AAPL), and Alaska Air Group Inc. (ALK), saw a general upward trend across the period—albeit, with some stagnation for ALK in the latter half of the decade. Higher volatility is seen around 2020, likely due to the COVID-19 pandemic.

The U.K. stock market, represented by the FTSE 100 index (FTSE100), Shell Plc (SHEL) and Taylor Wimpey Plc (TW), had mixed performance over the decade, with a noticeable 'V-shaped recovery' in 2015-2017, characterised by a sharp decline followed by a rapid recovery, likely due to the EU Referendum. Higher volatility is seen around 2020, likely due to the COVID-19 pandemic.

The Chinese stock market, represented by the SSE Composite index (SSE), Ping An Insurance Ltd (PING) and Shenzhen Salubris Pharmaceuticals Co Ltd (SHEN), had a strong performance over the decade with a significant bull run in 2015, driven by government encouragement of retail investing and some speculative trading (Sornette et al., 2015).

1.4 Conclusion

We have established motivation for reassessing financial models and the investment process to better account for uncertainty and tail risk—structuring this thesis around three objectives.

Objective One: Asset Returns Forecasting is addressed in Chapters 2, 3, and 4. Chapter 2 explores the analysis of financial time series data to understand market behaviours and patterns, which may be useful in developing more refined forecasting algorithms. Chapter 3 builds on this by examining various existing financial time series forecasting models. Chapter 4 explores the potential of Bayesian forecasting in enhancing the accuracy and reliability of financial predictions, advo-

cating for a shift towards models that account for market uncertainties. We present original research comparing methods for Bayesian evidence calculation in this context.

Objective Two: Risk Management is the focus of Chapters 5, 6, 7 and 8. Chapter 5 introduces quantitative financial risk measurement. Chapter 6 explores the application of Monte Carlo methods and importance sampling in risk analysis. In Chapter 7, advanced techniques for risk analysis are presented building to the presentation, in Chapter 8, of a new algorithm called the 'multicanonical sequential Monte Carlo sampler', which improves the ability to quantify the probability and magnitude of tail risk.

Objective Three: Optimal Portfolio Allocation is covered in Chapters 9 and 10. Chapter 9 reviews traditional portfolio optimisation algorithms and introduces a risk-based optimisation approach. Chapter 10 presents new Bayesian Optimisation algorithms for optimal portfolio allocation, including various demonstrative examples. Chapter 10 concludes with the results of our complete framework—price forecasting under uncertainty, tail risk management, and optimal portfolio allocation—combined and applied to historical data. Chapter 11 provides concluding remarks.

This thesis proposes a shift in financial modelling and investment decision-making. By analysing existing methodologies and proposing our own algorithms, we aim to provide a more realistic investment framework for navigating the uncertainties and tail risks of the financial markets.

Chapter 2

Financial Time Series Analysis

We have established the foundational principles of financial markets, emphasising the critical balance between risk and return for investors. Distinguishing risks that are measurable through probability from true uncertainty, where outcomes are unknown and not amenable to probabilistic prediction, is central to our first objective: to develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

Building forecasting models involves analysing a financial time series, such as historical stock returns data ($\{r_t\}_{t=1}^{t=T}$), to understand the relationship between the series value at time t and the information known before time t. This chapter aims to establish a foundation for time series analysis, paving the way for the presentation and development of forecasting models in subsequent chapters.

2.1 Time series analysis

The foundational concepts of normality and stationarity underpin many forecasting models—we briefly introduce them here, adapting the presentation of Tsay (2005).

2.1.1 Normality

A cornerstone of traditional financial models is the assumption of normality in financial returns, which presumes symmetry in the return distributions, where positive deviations from the mean are mirrored by negative ones—it is not only a statistical convenience; it fundamentally shapes how risk is measured.

However, the assumption is clearly flawed, as most financial time series and asset returns do not follow a Normal distribution (Bodie et al., 2002). For example, in our core data set, the methods of moments of the daily and monthly log returns (Tables 2.1) and the histograms plots (Plots 2.1 and 2.2)—which follow—indicate that the log returns time series do not follow a Normal distribution. Negative skewness indicates a pronounced left tail resulting from sporadic extreme negative returns. Positive excess kurtosis[1] indicates a heavy-tailed distribution and a more accentuated peak compared to a standard Normal distribution.

When analysing a financial time series and building forecasting models, it is imperative to check the assumption of normality. This can be achieved through the use of rigorous statistical tests, such as:

1. **Shapiro–Wilk test** examines the dataset to calculate a test statistic and corresponding p-value, assessing how closely the historical returns follow a Normal distribution.

2. **Anderson-Darling test** examines the fit of the data to the Normal distribution in terms of the order statistics of the dataset.

3. **Jarque-Bera test** concentrates on skewness and kurtosis.

4. **Kolmogorov-Smirnov test** compares the empirical cumulative distribution

[1]Kurtosis of 3 (*i.e.*, $K(x) = 3$) indicates a Normal distribution; as such, excess kurtosis is defined as $K(x) - 3$.

function of the data to the Normal distribution. Assessing the degree to which the data conforms to the expected distribution.

Table 2.1: Daily & Monthly Log Return Statistics

	Asset ticker	**Total dates**	**Mean** $(\times 10^{-2})$	**Std dev** $(\times 10^{-2})$	**Skewness**	**Kurtosis**	**Min value** $(\times 10^{-2})$	**Max value** $(\times 10^{-2})$
Daily	SP500	2778	0.04	1.10	−0.86	16.56	−12.77	8.97
	AAPL	2778	0.11	1.78	−0.30	6.54	−13.77	11.32
	ALK	2778	0.07	2.42	−0.78	16.24	−26.45	18.49
	FTSE100	2778	0.01	1.05	−0.67	9.94	−11.51	8.67
	SHEL	2778	0.01	1.66	−0.58	19.79	−19.35	18.55
	TW	2778	0.07	2.40	−1.08	19.62	−34.62	17.07
	SSE	2778	0.00	1.33	−0.88	6.64	−8.87	6.37
	PING	2778	0.04	1.88	0.04	5.41	−14.74	12.22
	SHEN	2778	0.04	2.44	−0.05	3.21	−10.56	9.54
Monthly	SP500	2758	0.91	4.41	−2.07	13.49	−40.00	22.35
	AAPL	2758	2.27	7.90	−0.40	0.64	−33.45	30.85
	ALK	2758	1.41	11.24	−1.98	14.61	−102.04	59.07
	FTSE100	2758	0.14	4.55	−2.06	13.47	−39.38	15.42
	SHEL	2758	0.25	7.53	−1.08	13.34	−67.89	40.87
	TW	2758	1.45	10.78	−1.41	7.94	−76.28	42.46
	SSE	2758	0.06	6.59	−0.38	2.93	−37.71	23.70
	PING	2758	0.92	9.05	−0.17	1.67	−48.81	42.64
	SHEN	2758	0.93	10.90	0.40	1.32	−31.50	51.34

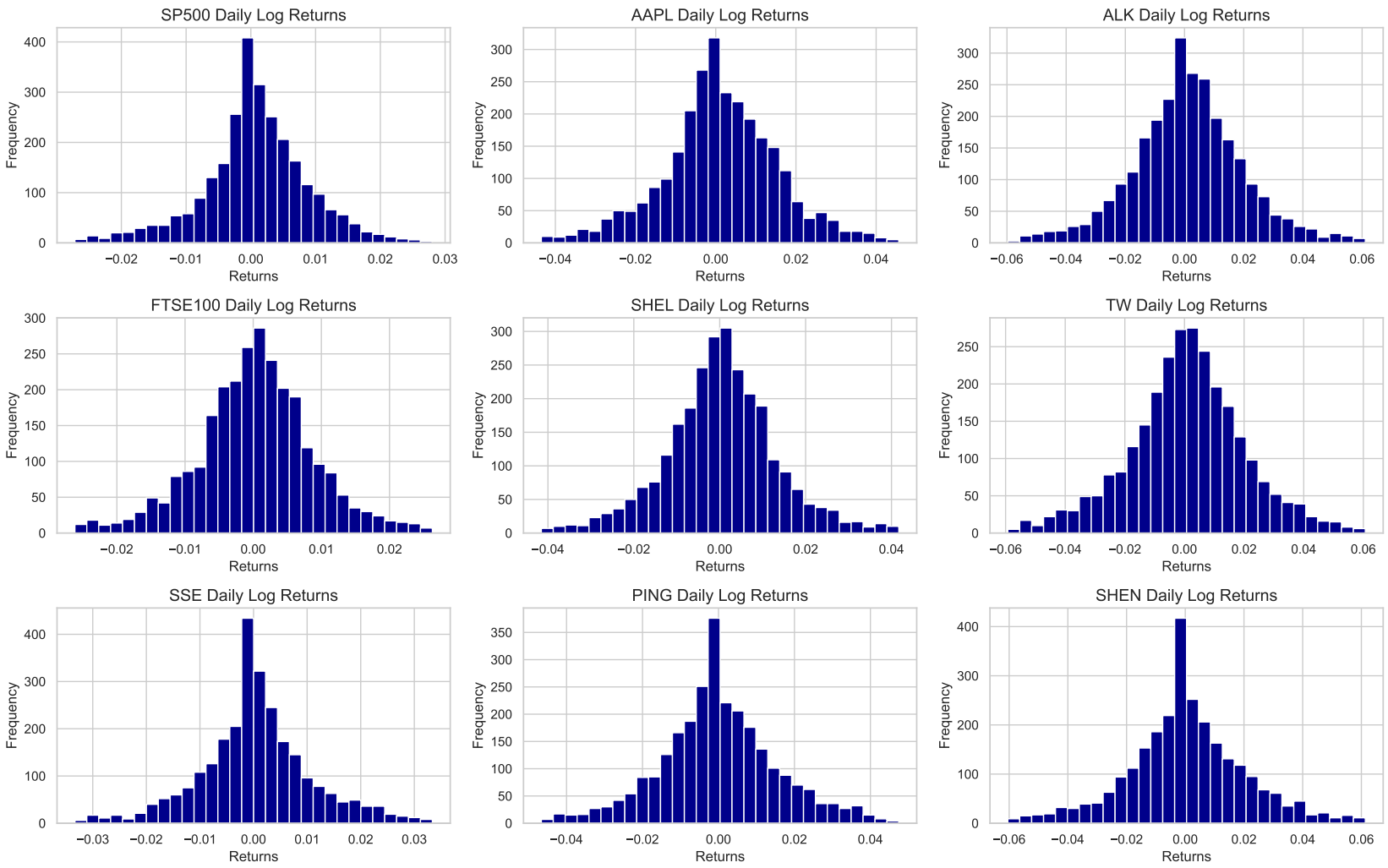


Figure 2.1: Histogram of historic daily log returns

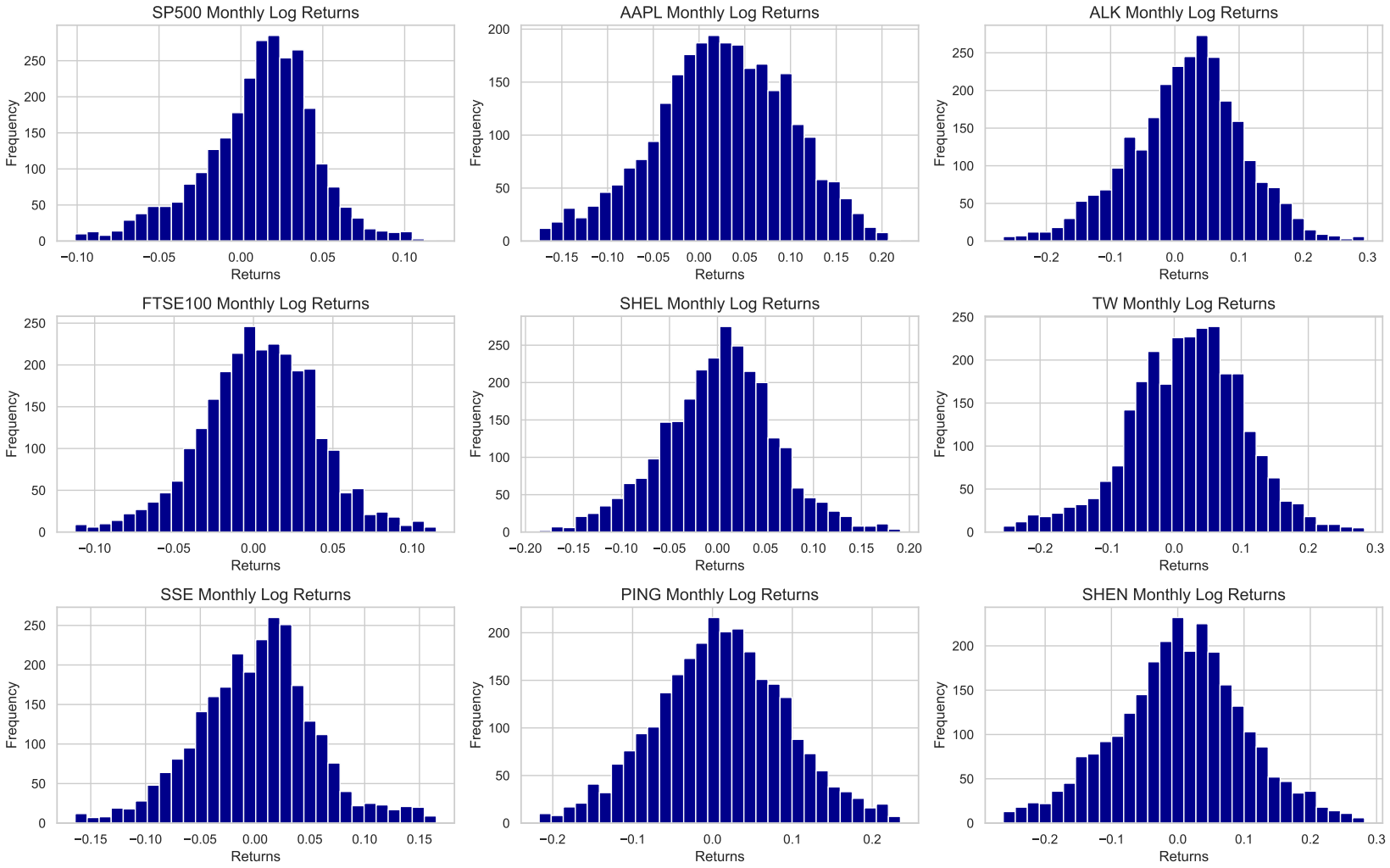


Figure 2.2: Histogram of historic monthly log returns

All four tests applied to our core data set concur that the historical asset returns are not Normally distributed[2]. Deviations from normality, especially when negative skewness is observed, can potentially understate extreme negative outcomes—or tail risk. Investors should move away from the normality assumption when analysing financial time series—a concept we return to in later sections.

2.1.2 Stationarity

We now focus on features of financial time series which may be leverageable in building price and volatility forecast models. The first feature is stationarity, which indicates that the statistical properties of a time series remain constant over time. Formally, a time series $\{r_t\}$ is *strictly stationary* if the joint distribution of $(r_{t_1}, ..., r_{t_k})$ is identical to that of $(r_{t_1+t}, ..., r_{t_k+t})$ for all t; where $(t_1, ..., t_k)$ is a collection of integers for any $k \in \mathbb{Z}^+$.

Clearly, strict stationarity is challenging to verify empirically; as such, a more practical, less stringent form of stationarity is often assumed—called *weak stationarity*. A time series is considered *weakly stationary* if for all t: (a) $\mathbb{E}(r_t) = \mu$, a constant mean, (b) $\text{Var}(r_t) = \sigma^2$, a constant variance, and (c) $\text{Cov}(r_t, r_{t-l}) = \gamma_l$, which depends solely on l, that is, the covariance between r_t and r_{t-l} is time-invariant. This covariance γ_l, called the lag-l autocovariance of r_t, holds two vital properties: (a) $\gamma_0 = \text{Var}(r_t)$ and (b) $\gamma_{-l} = \gamma_l$.

If a time series's statistical properties remain constant over time—where its values fluctuate around a consistent mean and follow a predictable pattern—it is considered stationary. The opposite of this would be a behaviour akin to a random walk, which does not revert to a constant level over time; therefore, the time series would have the presence of a unit root, as shown by Cheung and Lai (1995). Testing a time series for the presence of a unit root is easier than testing for stationarity.

[2]Detailed results are in Appendix A.1.

Augmented Dickey–Fuller test

The Augmented Dickey–Fuller (ADF) test is used to establish whether a time series exhibits stationarity by testing for the presence of a unit root. The test's null hypothesis is that the time series contains a unit root, implying non-stationarity; the alternative hypothesis is that the time series exhibits stationarity. To demonstrate this test, we applied it to our key data across the whole period from January 1, 2010 to December 30, 2020—with a significance level set at $\alpha = 0.05$.

Asset	**p-value**	**Stationary?**
SP500	1.73×10^{-27}	Yes
AAPL	4.80×10^{-30}	Yes
ALK	1.17×10^{-17}	Yes
FTSE100	2.57×10^{-30}	Yes
SHEL	3.59×10^{-20}	Yes
TW	0.0	Yes
SSE	5.06×10^{-16}	Yes
PING	0.0	Yes
SHEN	9.31×10^{-26}	Yes

(a) Daily log returns

Asset	**p-value**	**Stationary?**
SP500	9.25×10^{-12}	Yes
AAPL	2.20×10^{-10}	Yes
ALK	1.57×10^{-10}	Yes
FTSE100	1.40×10^{-10}	Yes
SHEL	2.99×10^{-09}	Yes
TW	8.71×10^{-1}	Yes
SSE	2.01×10^{-08}	Yes
PING	9.02×10^{-11}	Yes
SHEN	9.94×10^{-11}	Yes

(b) Monthly log returns

Table 2.2: ADF test results

The results in Tables 2.2a and 2.2b indicate that the time series relating to our core data exhibit stationarity over the whole period.

Applying the ADF test for each year separately, rather than over the whole period, indicates that the daily log returns exhibit stationarity across all years—the results are given in Appendix A.2. In contrast, the stationarity of monthly log returns varies by year (Table 2.3).

A time series that exhibits stationarity has stable and predictable statistical properties through time, which can be utilised in forecasting models. In contrast, building a model on a non-stationary time series is very challenging—as one cannot

utilise such properties; leading to unreliable and spurious results when traditional statistical analysis is applied.

Asset	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Portfolio	✓	✓	✓	✓	×	✓	✓	×	×	✓	✓
SP500	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
AAPL	✓	✓	×	✓	✓	×	✓	✓	×	✓	×
ALK	✓	×	✓	×	✓	✓	×	✓	✓	✓	✓
FTSE100	✓	✓	✓	✓	✓	✓	×	✓	×	✓	✓
SHEL	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TW	×	✓	×	✓	✓	✓	✓	×	✓	×	✓
SSE	✓	×	×	✓	×	×	✓	✓	✓	✓	✓
PING	×	✓	✓	×	×	×	×	✓	✓	✓	×
SHEN	×	✓	✓	✓	×	✓	✓	×	✓	✓	×

Table 2.3: Stationarity status of monthly log returns by year. ✓ denotes stationarity. × denotes non-stationarity.

2.2 Autocorrelation

By understanding the relationship between an asset return r_t at time t and the information known before time t, it is possible to build forecasting models on a stationary time series $\{r_t\}$ knowing that such a relationship is stable over time.

One route to do this, is to consider the correlation between the asset return r_t and its historical values, referred to as *autocorrelation* or *serial correlation*; a distance measure of how closely current asset returns relate to their past values.

For a weakly stationary return series r_t, the linear dependence between r_t and its past values r_{t-l} is termed the lag-l *autocorrelation* of r_t, denoted by p_l. For example, in daily stock returns, a lag-1 autocorrelation represents the correlation between today's return and yesterday's. The autocorrelation coefficient for a weakly stationary time series at lag-l is defined by Tsay (2005) as:

$$p_l = \frac{\mathrm{Cov}(r_t, r_{t-l})}{\sqrt{\mathrm{Var}(r_t)\,\mathrm{Var}(r_{t-l})}} = \frac{\mathrm{Cov}(r_t, r_{t-l})}{\mathrm{Var}(r_t)} = \frac{\gamma_l}{\gamma_0}. \tag{2.2.1}$$

We make several immediate observations:

1. $p_0 = 1$: The autocorrelation of a series with itself (lag-0) is always one; as any series is perfectly correlated with itself.

2. **Symmetry,** $p_l = p_{-l}$: Autocorrelation is symmetric around lag-0, meaning the correlation at lag l is the same as at lag $-l$.

3. **Range,** $-1 \leqslant p_l \leqslant 1$: An autocorrelation coefficient of -1, 0 or 1, respectively, indicates a perfect negative, zero, or a perfect positive linear correlation.

4. A weakly stationary series r_t is not serially correlated if and only if $p_l = 0$ for all $l > 0$.

By assessing the autocorrelation of a time series, one can determine its memory and predictability. Autocorrelation provides a basis for understanding how past values might influence future values; crucial in financial modelling and forecasting.

2.2.1 Ljung–Box test

The opposite of a serially correlated time series is a white noise sequence, defined as a collection of independent identically distributed random variables with constant mean and variance but no autocorrelation between data points. While directly testing for serial correlation is hard, testing for white noise is relatively straightforward.

The -Box statistical test, introduced by Ljung and Box (1978), applied to time series, has a null hypothesis that the series is a white noise process. Conversely, a significant p-value indicates the presence of serial correlation. The Ljung–Box test statistic (Q) is given by:

$$Q = n(n+2)\sum_{l=1}^{h}\frac{p_l^2}{n-l}, \tag{2.2.2}$$

for sample size n, number of lags being tested h, and sample autocorrelation p_l at lag l. Under the null hypothesis (no autocorrelation) and when certain regularity conditions are satisfied, the test statistic Q follows a χ^2 distribution with h degrees of freedom. These conditions include the independence of observations, normality of residuals, and the assumption of no autocorrelation at lag lengths beyond those tested.

We applied the Ljung–Box test to our core daily and monthly log returns data, with lag periods of $5, 10, 20$ and 30 for daily returns and $1, 6, 12$ and 24 for monthly returns. For the purpose of this test, we computed the monthly log returns for each calendar month in isolation, to ensure no overlap between the periods which would inadvertently cause autocorrelation.

For the daily log returns, the Ljung–Box tests indicate the presence of significant serial correlation for all assets across all tested lags except the FTSE 100 and PING. The FTSE 100 shows no significant correlation at lag 5 but demonstrates significance at higher lag values. Conversely, PING has no significant serial correlation at lags 10 and 20 but significance at lags 5 and 30. For the monthly log returns, the Ljung–Box results show no significant serial correlation at the 5% level for any assets. The historical monthly returns are unlikely to be useful in indicating future values. In contrast, the daily log returns exhibit significant autocorrelation—so past returns may be useful in predicting future returns.

Daily log returns

Symbol	Lag	LB statistic	p-value	Serial correlation
SP500	5	103.8	8.28e-21	Yes
	10	238.39	1.49e-45	Yes
	20	292.91	2.27e-50	Yes
	30	308.43	5.76e-48	Yes
AAPL	5	11.18	4.79e-2	Yes
	10	44.12	3.13e-6	Yes
	20	66.31	7.20e-7	Yes
	30	75.48	8.63e-6	Yes
ALK	5	14.43	1.31e-2	Yes
	10	47.59	7.37e-7	Yes
	20	98.64	2.21e-12	Yes
	30	115.12	6.64e-12	Yes
FTSE100	5	3.42	6.36e-1	No
	10	34.62	1.4e-4	Yes
	20	54.96	4.17e-5	Yes
	30	70.11	4.70e-5	Yes
SHEL	5	19.38	1.61e-3	Yes
	10	47.51	7.62e-7	Yes
	20	79.17	5.44e-9	Yes
	30	99.08	2.59e-9	Yes
TW	5	18.94	1.97e-3	Yes
	10	22.68	1.19e-2	Yes
	20	37.69	9.66e-3	Yes
	30	64.79	2.33e-4	Yes
SSE	5	12.73	2.6e-2	Yes
	10	42.26	6.74e-6	Yes
	20	67.24	5.11e-7	Yes
	30	111.67	2.45e-11	Yes
PING	5	12.95	2.38e-2	Yes
	10	17.82	5.81e-2	No
	20	25.91	1.68e-1	No
	30	44.59	4.21e-2	Yes
SHEN	5	32.43	4.88e-6	Yes
	10	40.61	1.32e-5	Yes
	20	65.30	1.04e-6	Yes
	30	90.77	5.02e-8	Yes

Table 2.4: Ljung–Box test results for daily log returns

Monthly log returns

Symbol	Lag	LB statistic	p-value	Serial correlation
SP500	1	2.35	1.2e-1	No
	6	5.81	4.44e-1	No
	12	8.17	7.72e-1	No
	24	26.11	3.45e-1	No
AAPL	1	1.01	3.14e-1	No
	6	6.38	3.82e-1	No
	12	12.69	3.92e-1	No
	24	23.72	4.78e-1	No
ALK	1	0.09	7.60e-1	No
	6	1.23	9.75e-1	No
	12	5.96	9.18e-1	No
	24	14.49	9.34e-1	No
FTSE100	1	0.16	6.91e-1	No
	6	1.34	9.69e-1	No
	12	6.46	8.91e-1	No
	24	12.38	9.75e-1	No
SHEL	1	1.12	2.92e-1	No
	6	6.66	3.54e-1	No
	12	10.69	5.56e-1	No
	24	13.55	9.56e-1	No
TW	1	0.32	5.72e-1	No
	6	1.61	9.52e-1	No
	12	8.18	7.71e-1	No
	24	13.79	9.51e-1	No
SSE	1	0.67	4.12e-1	No
	6	4.34	6.31e-1	No
	12	15.60	2.10e-1	No
	24	26.63	3.22e-1	No
PING	1	0.15	7.02e-1	No
	6	4.58	5.98e-1	No
	12	10.99	5.29e-1	No
	24	18.57	7.75e-1	No
SHEN	1	3.02	8.2e-2	No
	6	5.73	4.55e-1	No
	12	10.22	5.97e-1	No
	24	23.38	4.98e-1	No

Table 2.5: Ljung–Box test results for monthly log returns

2.2.2 Autocorrelation plots

The daily log returns exhibit serial correlation, implying that past values can help predict future values. Autocorrelation function (ACF) plots and partial autocorrelation function (PACF) plots can help determine the significant lags to capture this correlation.

ACF plots show the autocorrelation between a time series and its lagged values at different time lags. PACF plots show the partial autocorrelation between a time series and its lagged values, controlling for the influence of intermediate lag. Within ACF and PACF plots, the correlations are plotted along with confidence intervals, representing the range within which the values would fall under the assumption of no autocorrelation (white noise). ACF and PACF values outside of these confidence intervals indicate significant lags.

We generated ACF and PACF plots for our core daily log returns data. As the underlying daily log returns are not normally distributed, we used bootstrapping[3] to obtain a more accurate estimation of the confidence intervals without the need for normality assumptions. The ACF and PACF plots are shown below for lag values up to 42, with a bootstrapping sample size of 1000 and 95% confidence intervals.

[3]Bootstrapping is a resampling technique used to estimate the sampling distribution of data, which repeatedly resamples from the observed data with replacement.

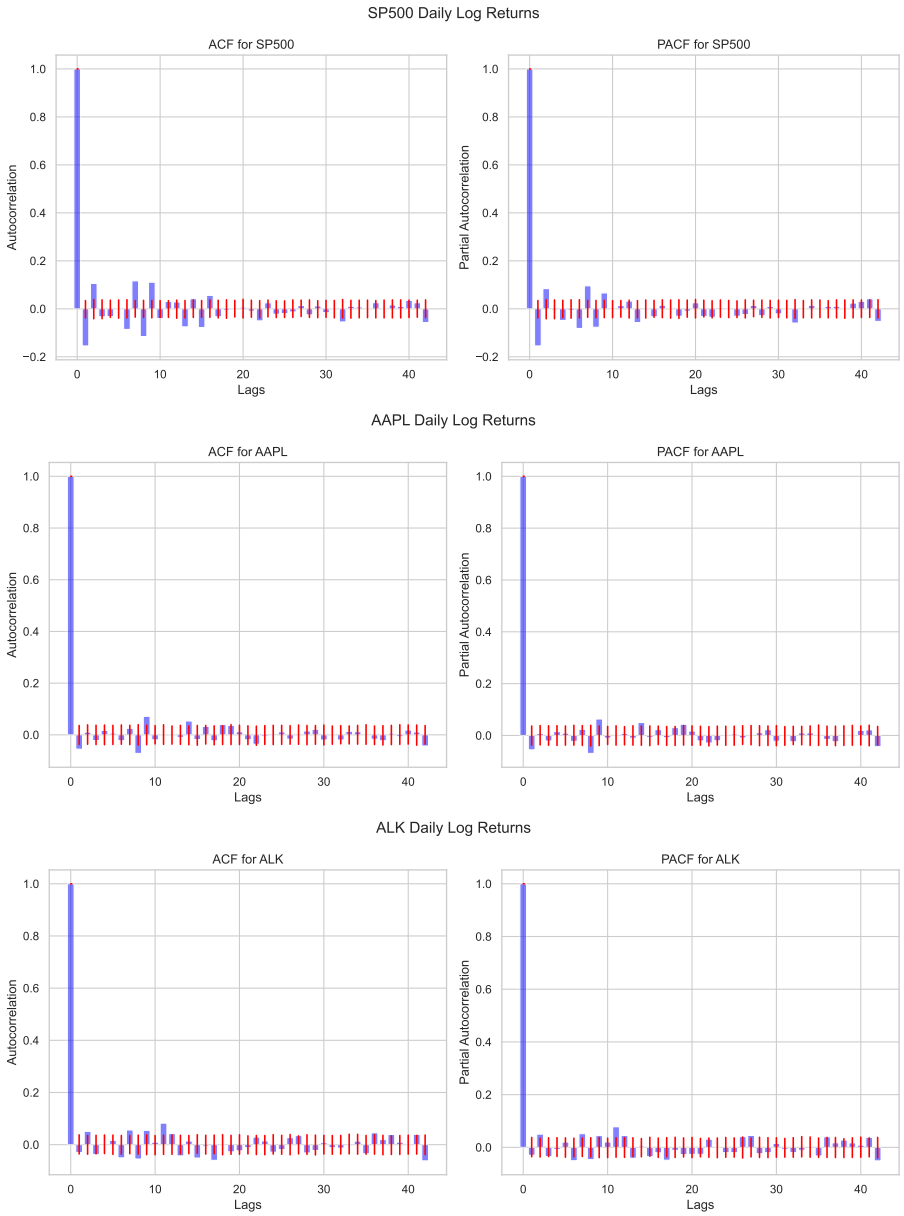


Figure 2.3: Autocorrelation and partial autocorrelation plots of daily log returns (US)

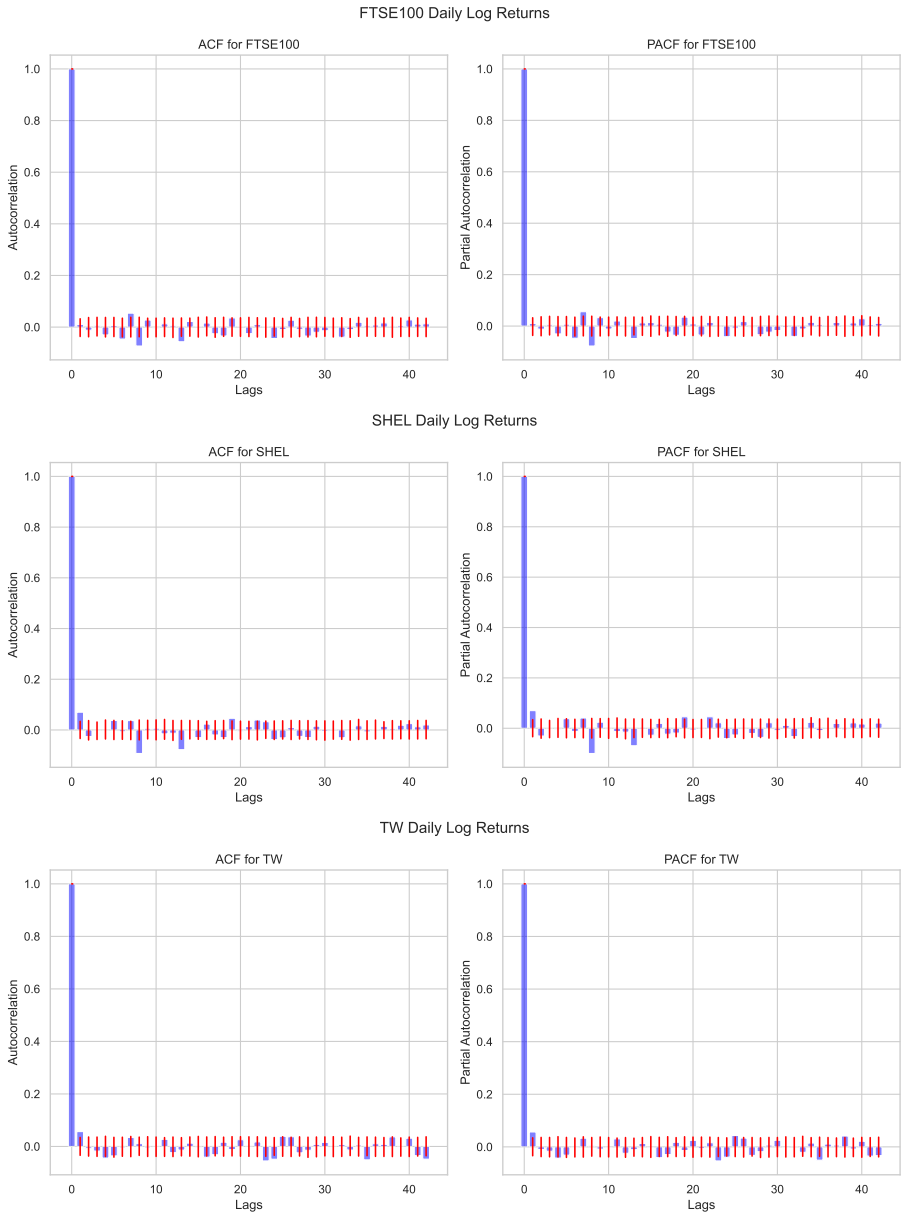


Figure 2.4: Autocorrelation and partial autocorrelation plots of daily log returns (UK)

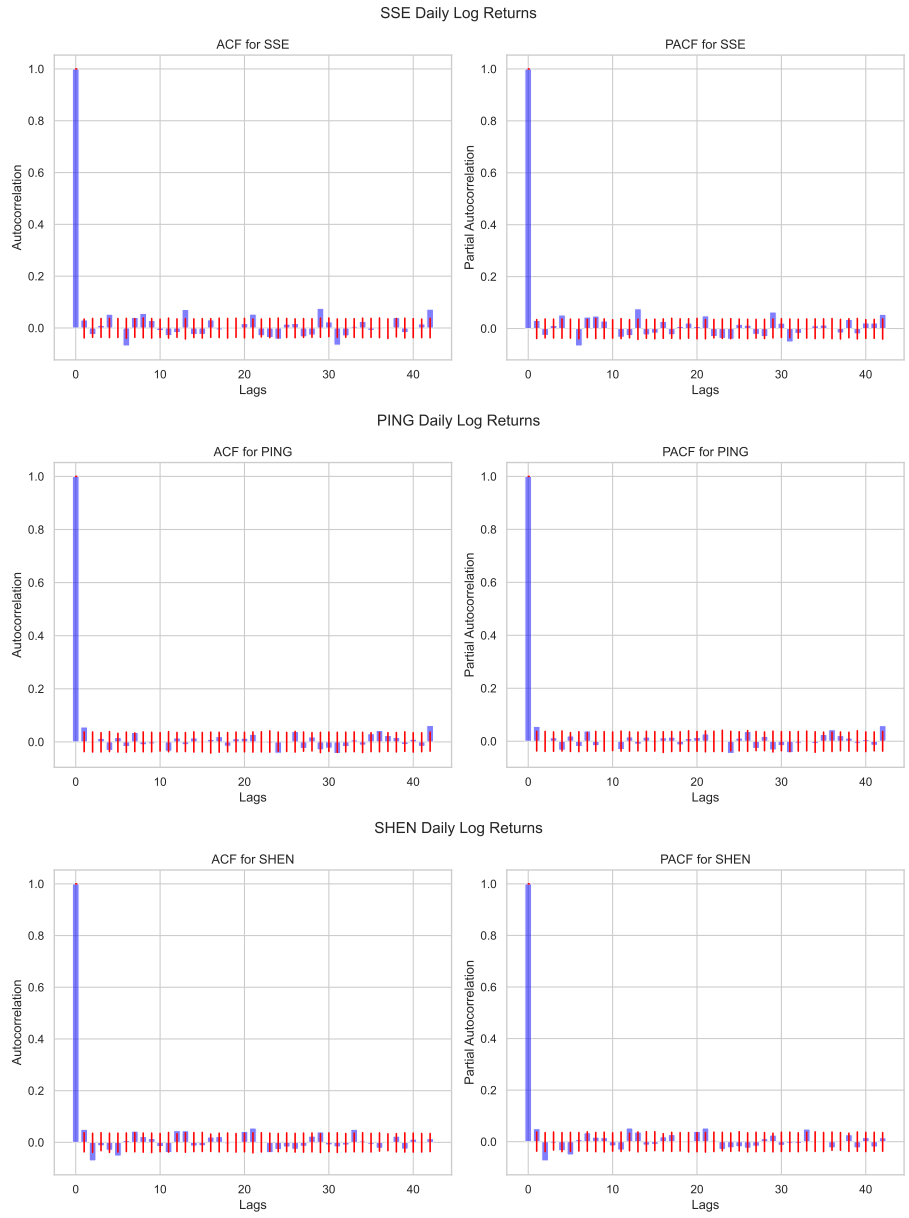


Figure 2.5: Autocorrelation and partial autocorrelation plots of daily log returns (CN)

The significant lag values falling outside the confidence intervals are listed in Table 2.6.

Asset ticker	**Significant ACF lags**	**Significant PACF lags**
SP500	1, 2, 6, 7, 8, 9, 10, 13, 14, 15, 16, 22	1, 2, 4, 6, 7, 8, 9, 13, 15
AAPL	1, 8, 9, 14, 18, 19, 22	1, 8, 9, 14, 19
ALK	2, 6, 7, 8, 9, 11, 12, 13, 15, 17	2, 6, 7, 8, 9, 11, 12, 13, 17
FTSE100	6, 7, 8, 13, 24	6, 7, 8, 13, 18, 24
SHEL	1, 7, 8, 13, 19, 22, 24	1, 5, 7, 8, 13, 19, 22, 23
TW	1, 4, 16, 23	1, 4, 16, 23
SSE	4, 6, 7, 8, 13, 21, 23	4, 6, 7, 8, 13, 21, 23
PING	1, 7, 23	1, 7, 23
SHEN	1, 2, 5, 7, 12, 13, 20, 21, 23	1, 2, 5, 12, 13, 20, 21

Table 2.6: Significant autocorrelation and partial autocorrelation lags.

Serial correlation suggests a departure from the random walk hypothesis often assumed in financial markets, thereby providing a foundation for predictive modelling. For our core daily log returns data, significant p-values at various lags indicate the presence of serial correlation, suggesting that past values influence future values. This finding is pivotal, as it provides a route to building forecasting methods.

2.3 Time series summary

In conclusion, this chapter has explored the concepts of normality, stationarity, and autocorrelation within time series analysis, setting the stage for the development of forecasting models. Here's how each aspect contributes to the overarching goal of developing advanced forecasting methods:

Normality: The assumption of normality in financial data underpins many traditional forecasting models, despite it being widely accepted that financial data is non-Normal. As such, new mathematical methods should remove the normality assumption and better reflect the true probability distribution of financial data.

Stationarity: Forecasting models must consider the stationarity (or lack thereof) in a financial time series. The presence of stationarity ensures statistical properties remain constant over time allowing them to be exploited for forecasting purposes.

Autocorrelation: Autocorrelation challenges the random walk hypothesis and indicates that past values might inform future ones. Recognising and leveraging autocorrelation in financial data can lead to more insightful and accurate forecasting models, especially in capturing temporal dependencies.

The first objective of this thesis is to develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty. By leveraging the analysis from this chapter, we have a route to develop such forecasting methods.

Chapter 3

Financial Time Series Forecasting

Weak stationarity of a time series implies stability in the statistical properties over time. This, combined with autocorrelation, posits that past returns may help forecast future returns. In this chapter, we introduce forecasting models that take advantage of autocorrelation, dividing this chapter into two parts: return forecasting and volatility forecasting. By understanding these models and their limitations, we motivate the development of new return and volatility forecasting models, which better account for uncertainty.

3.1 Return forecasting models

Return forecasting models utilise the relationship between past and current return data to forecast future returns. The autoregressive model (AR) leverages the relationship between a variable and its lags; the moving average model (MA) focuses on past forecast errors; and the autoregressive moving average model (ARMA) combines the two. We now introduce these models, their underlying assumptions, and implementation details. The work in this section and the next draws on the presen-

tation by Tsay (2005).

3.1.1 Linearity

A fundamental consideration in time series analysis is whether the time series r_t exhibits linear behaviour, where r_t can be expressed as a linear function of its past values and some shocks or errors. Shocks refer to random and unpredictable events that cause significant deviations in the time series. In this context, errors are the deviations of observed values from the predicted values of the model, encapsulating the impact of these unpredictable shocks. Therefore, while shocks represent the actual unforeseen events affecting the time series, errors represent the discrepancies between the model's forecasts and the actual values, often influenced by such shocks.

Formally, r_t is linear if it can be written as (Tsay, 2005):

$$r_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i}, \tag{3.1.1}$$

where the first component is the (constant) mean of the series μ and the second component is a summation across various lags of past random *shocks* a_{t-i} weighted by their importance ψ_i. The shocks are represented by $\{a_{t-i}\}_{i=0}^{\infty}$, a sequence of independent and identically distributed (i.i.d.) random variables embodying white noise with mean zero; implying that the time series is influenced by some random, unpredictable factors, distinguishing linear time series from more deterministic patterns.

This thesis considers the case where a_t is a continuous random variable and the asset returns r_t exhibit weak stationarity. As such, the time series' mean and variance are:

$$\mathbb{E}(r_t) = \mu, \quad \text{Var}(r_t) = \sigma_a^2 \sum_{i=0}^{\infty} \psi_i^2, \tag{3.1.2}$$

where σ_a^2 represents the variance of a_t.

3.1.2 Autoregressive model

The first return forecasting model we explore is the simple autoregressive model, which captures the influence of past values on future values. If a time series can be expressed as a linear function of its past data, we focus on utilising autocorrelation. By identifying significant autocorrelation across different lag periods, $p \in \mathbb{Z}^+$, we know that the lagged return will likely be useful in predicting r_t. To take advantage of this, one can define a simple autoregressive model AR(p) for $p \in \mathbb{Z}^+$ as:

$$r_t = \phi_0 + \phi_1 r_{t-1} + ... + \phi_p r_{t-p} + a_t, \tag{3.1.3}$$

where a_t is a white noise series with mean zero and variance σ_a^2; and ϕ_i represent the autoregressive parameters to be tuned.

The AR(p) model posits that the previous p values, denoted as r_{t-i} (for $i = 1, ..., p$), collectively influence the conditional expectation of r_t. This model structure resembles a multiple linear regression model, albeit with a crucial difference: the independent variables are the lagged values from the same time series.

The optimal order p is typically not known in advance and needs to be determined through empirical analysis. This process is known as order determination and has been the subject of extensive research. There are two primary methods: the first utilises the partial autocorrelation function, while the second employs specific information criterion functions. We will return to this discussion shortly.

Expectation of r_t

The expectation of r_t under the AR(p) model is

$$\begin{aligned}\mathbb{E}(r_t) &= \mathbb{E}(\phi_0 + \phi_1 r_{t-1} + \cdots + \phi_p r_{t-p} + a_t) \\ &= \phi_0 + \phi_1 \,\mathbb{E}(r_{t-1}) + \cdots + \phi_p \,\mathbb{E}(r_{t-p}),\end{aligned}$$

using the linearity of expectation and given $\mathbb{E}(a_t) = 0$. Now assuming stationarity, where $\mathbb{E}(r_{t-i}) = \mu$ for all i, by deriving

$$\mu = \frac{\phi_0}{1 - (\phi_1 + \cdots + \phi_p)}, \tag{3.1.4}$$

one can understand how the model's parameters influence the mean of the series.

Variance of r_t

Assuming stationarity and $\mathrm{Var}(a_t) = \sigma_a^2$, then the variance of r_t is:

$$\mathrm{Var}(r_t) = \phi_1^2 \,\mathrm{Var}(r_{t-1}) + \cdots + \phi_p^2 \,\mathrm{Var}(r_{t-p}) + \sigma_a^2. \tag{3.1.5}$$

Given that $\mathrm{Var}(r_{t-i}) = \sigma^2$ for all i under stationarity, this simplifies to:

$$\sigma^2 = \frac{\sigma_a^2}{1 - \sum_{i=1}^{p} \phi_i^2}. \tag{3.1.6}$$

This captures the cumulative impact of the autoregressive terms on the overall variance of the time series (σ^2).

Forecasting

Having established an AR(p) model for a financial time series, it can be used to forecast the value of $r_{h+\ell}$ with *forecast horizon* $\ell \geqslant 1$ starting at *forecast origin* h.

The forecast of $r_{h+\ell}$ is denoted by $\widehat{r}_h(\ell)$ and utilises all available data up to the forecast origin, represented by F_h. The l-step ahead forecast $\widehat{r}_h(\ell)$ is selected such that:

$$\mathbb{E}\left[\left(r_{h+\ell} - \widehat{r}_h(\ell)\right)^2 | F_h\right] \leqslant \min_g \mathbb{E}\left[\left(r_{h+\ell} - g\right)^2 | F_h\right], \tag{3.1.7}$$

where g is a function of F_h that encapsulates the information available up to time h. For a general AR(p) model and using a minimum squared error loss function, the ℓ-step ahead forecast can be considered as the conditional expectation of $r_{h+\ell}$ given F_h:

$$\widehat{r}_h(\ell) = \phi_0 + \sum_{i=1}^{p} \phi_i \widehat{r}_h(\ell - i), \tag{3.1.8}$$

where $\widehat{r}_h(i) = r_{h+i}$ if $i \leqslant 0$. This allows iterative computation of the forecast $\widehat{r}_h(i)$ for $i = 1, \ldots, \ell - 1$.

3.1.3 Simple moving average

We now explore the moving average (MA) model, which focuses on the impact of past forecast errors rather than past series values. Unlike the autoregressive (AR) model, which incorporates the influence of past values and the shocks affecting them, the MA model emphasises the deviations (errors) from previous forecasts.

We introduce MA models by considering them as an infinite-order AR model with some parameter constraints, following the approach of Tsay (2005). In theory, a time series $\{r_t\}$ can be modelled using an AR model with infinite order of the form:

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + ... + a_t. \tag{3.1.9}$$

However, this model is not feasible since it requires infinite parameters. To make the model more realistic, constraints can be imposed on the coefficients ϕ_t, reducing

them to a finite number of parameters:

$$r_t = \phi_0 - \theta_1 r_{t-1} - \theta_1^2 r_{t-2} - \theta_1^3 r_{t-3} - ... + a_t, \tag{3.1.10}$$

where the coefficients depend on the single parameter θ_1 via $\phi_i = -(\theta_1)^i$ for $i \geqslant 1$. Writing this in a compact form:

$$r_t + \theta_1 r_{t-1} + \theta_0^2 r_{t-2} + ... = \phi_0 + a_t, \tag{3.1.11}$$

from which the model for r_{t-1} is:

$$r_{t-1} + \theta_1 r_{t-2} + \theta_1^2 r_{t-3} + ... = \phi_0 + a_{t-1}. \tag{3.1.12}$$

Multiplying Eq. 3.1.12 by θ_1 and subtracting the result from Eq. 3.1.11, obtains:

$$r_t = \phi_0(1 - \theta_1) + a_t - \theta_1 a_{t-1}. \tag{3.1.13}$$

In the resulting equation, the current value r_t is influenced by a combination of components: a constant term ϕ_0 multiplied by one minus the coefficient θ_1; the present shock or error term a_t; and the previous period's shock a_{t-1} weighted by the coefficient θ_1. As such, r_t can be considered as a weighted average of the immediate unexpected change with a fraction of the preceding periods' shocks[1]. The general form MA(q) is:

$$r_t = c_0 + a_t - \theta_1 a_{t-1} - ... - \theta_q a_{t-q}, \tag{3.1.14}$$

for a non-negative integer q, a constant c_0 and white noise series a_t.

[1]In this context, the term "error" refers to the deviations in the model's predictions, often caused by unpredictable shocks. Therefore, "shock" and "error" are used interchangeably to reflect the impact of unforeseen events on the time series.

The optimal order q is typically not known in advance and needs to be determined through empirical analysis. An ACF can be used to identify the appropriate order of the MA model, known by q, or other more advanced methods, explored later.

Expectation of r_t

To determine the expectation of r_t under the MA(q) model, consider that

$$\mathbb{E}[r_t] = \mathbb{E}[c_0 + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}]. \tag{3.1.15}$$

Given that a_t is white noise with zero mean, we obtain

$$\mathbb{E}[a_t] = \mathbb{E}[a_{t-1}] = \cdots = \mathbb{E}[a_{t-q}] = 0 \tag{3.1.16}$$

Thus,

$$\mathbb{E}[r_t] = \mathbb{E}[c_0] + \mathbb{E}[a_t] - \theta_1 \mathbb{E}[a_{t-1}] - \cdots - \theta_q \mathbb{E}[a_{t-q}] \tag{3.1.17}$$

$$= c_0. \tag{3.1.18}$$

As such, the expected value of the time series r_t is equal to the constant term c_0, so it is uninfluenced by the white noise components.

Variance of r_t

To find the variance of r_t, we obtain:

$$\mathrm{Var}(r_t) = \mathrm{Var}(a_t) + \theta_1^2 \mathrm{Var}(a_{t-1}) + \cdots + \theta_q^2 \mathrm{Var}(a_{t-q}) \tag{3.1.19}$$

and given that each a_i term has a variance σ_α^2,

$$\mathrm{Var}(r_t) = \sigma_\alpha^2(1 + \theta_1^2 + \cdots + \theta_q^2). \tag{3.1.20}$$

Therefore, as MA models are a finite linear combination of a white noise sequence for which the first two moments are time-invariant, they can be considered to exhibit weak stationarity (Tsay, 2005).

Forecasting

As the MA(q) model only considers the previous q white noise terms, the effect of a shock to r_t will persist for q periods, after which its influence will disappear entirely. For a forecast horizon h, the forecasted value will converge to the series' mean.

For instance, in an MA(1) model, a shock at time t will influence forecasts for r_{t+1}, but when forecasting r_{t+2} and beyond, the shock's effect vanishes. This swift convergence to the mean is a distinctive feature where the multistep ahead forecasts converge to the series mean. The MA(q) for the period $h + l$ is defined by:

$$r_{h+l} = c_0 + a_{h+l} - \sum_{j=1}^{q} \theta_j a_{h+l-j}. \tag{3.1.21}$$

So, the value at time $h+l$ is influenced by a constant term c_0, the error term for the period a_{h+l}, and the error terms from the previous q periods, each being multiplied by their respective θ parameters.

By evaluating the conditional expectation of the forecast, based on the information available at time h, an l-step ahead forecast $\widehat{r}_h(l)$ can be obtained using:

$$\widehat{r}_h(l) = \mathbb{E}(r_{h+l}|F_h) = c_0 - \sum_{j=1}^{q} \theta_j a_{h+l-j}. \tag{3.1.22}$$

The forecast error, $e_h(l)$, the difference between the true value and the predicted value, is calculated using:

$$e_h(l) = r_{h+l} - \widehat{r}_h(l). \tag{3.1.23}$$

Understanding the variance associated with this forecast error is essential in practical

contexts. In the MA(q) framework, the variance of the l-step ahead forecast error can be generalised as:

$$\text{Var}[e_h(l)] = \sigma^2 \left(1 + \sum_{j=1}^{q} \theta_j^2\right). \tag{3.1.24}$$

This is equal to the variance of the series; as such, for a stationary series, the multistep ahead forecasts converge both to the series mean and the forecast errors to the variance of the series.

3.1.4 Autoregressive moving average

With its focus on previous time series values, the AR model excels in capturing the persistence of shocks over time. However, it may fall short in scenarios where the impact of a shock is not solely dependent on the series's own past values. On the other hand, the MA model accounts for random errors, which are often caused by unpredictable shocks impacting the time series, but has limited memory, only incorporating recent disturbances and neglecting the more extended historical context.

This is where the combined strength of the autoregressive moving average (ARMA) model offers a better framework, where past values (AR component) and past error terms (MA component) jointly influence the present value. A general ARMA(p, q) model is represented by (Tsay, 2005):

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + a_t - \sum_{j=1}^{q} \theta_j a_{t-j}, \tag{3.1.25}$$

where a_t is a white noise series, r_{t-i} are past series values, and ϕ_i and θ_j represent parameters to be tuned. The model orders p and q are non-negative integers. We note that there are no common factors between these two polynomials; otherwise, it is possible to reduce the model's order (p, q).

Forecasting

Let F_h denote the information set available at the forecast origin time h. This information set F_h includes all past observations of the time series up to and including time h, such as past returns and error terms.

The l-step ahead forecast of r_{h+l} using an ARMA model is given by (Tsay, 2005):

$$\widehat{r}_h(\ell) = \mathbb{E}(r_{h+\ell} \mid F_h) = \phi_0 + \sum_{i=1}^{p} \phi_i \widehat{r}_{h+\ell-i} - \sum_{j=1}^{q} \theta_j a_{h+\ell-j}. \tag{3.1.26}$$

Here, $\mathbb{E}(r_{h+\ell} \mid F_h)$ represents the conditional expectation of the future value $r_{h+\ell}$ given the information set F_h containing all available past information up to time h. This expectation is computed based on the ARMA model, which includes the parameters ϕ_0, ϕ_i (for $i = 1, 2, \ldots, p$), and θ_j (for $j = 1, 2, \ldots, q$).

Multistep-ahead forecasts using the ARMA model are generated in a stepwise manner, where each successive forecast depends on the previously forecasted values and errors.

ARMA Models

With their integrated approach, ARMA models offer advantages over simpler models by efficiently capturing the linear aspects of time series data and leveraging past values and error terms, thereby providing a more comprehensive view of market dynamics.

Financial markets are notorious for their unpredictable and often volatile nature, which ARMA models, in their standard form, do not directly address. This limitation is particularly pronounced in periods of market turbulence, where volatility tends to cluster, leading to large swings in asset prices. The inability of ARMA models to account for such changing volatility can lead to less reliable forecasts, especially in these turbulent periods.

3.2 Volatility forecasting models

Modelling volatility presents a unique challenge, as it is not directly observable; however, certain consistent characteristics can be taken advantage of:

1. **Volatility clustering**: Periods of high or low volatility tend to cluster together in financial time series.

2. **The leverage effect**: Volatility reacts differently to significant price movements, where volatility tends to increase more when asset prices decrease.

One class of volatility models which take advantage of these features are conditional heteroscedasticity models. Conditional heteroscedasticity implies that the variance of a time series is not constant over time but varies depending on certain conditions. We now introduce such models, drawing on the presentation by Tsay (2005).

3.2.1 Background

Given a data set accessible at time $t-1$, denoted by the information set F_{t-1}, which includes all historical values and error terms up to time $t-1$, an ARMA model can be used to obtain the mean and variance of r_t conditioned on F_{t-1}. This is defined as follows:

$$\mu_t = \mathbb{E}(r_t|F_{t-1}), \tag{3.2.1}$$

$$\sigma_t^2 = \mathrm{Var}(r_t|F_{t-1}) = \mathbb{E}((r_t - \mu_t)^2|F_{t-1}). \tag{3.2.2}$$

In this context, $\mu_t = \mathbb{E}(r_t|F_{t-1})$ represents the conditional expectation of r_t, which is the expected value of the return at time t given all the information available up to time $t-1$. This expectation is derived using an ARMA model, which captures the linear relationship between the current return and past returns, as well as past

error terms.

Similarly, $\sigma_t^2 = \text{Var}(r_t|F_{t-1}) = \mathbb{E}((r_t - \mu_t)^2|F_{t-1})$ represents the conditional variance of r_t. It is the expected value of the squared deviation of r_t from its mean, μ_t, given the information set F_{t-1}. This variance reflects the uncertainty or volatility of the return at time t based on past information.

Therefore, the time series r_t can be decomposed into two components: $r_t = \mu_t + a_t$, where μ_t represents the predictable component based on past information (using an ARMA model), and a_t is the unpredictable error term or residual. The residual term a_t is crucial as it embodies the uncertainty in the time series not captured by the ARMA model. It reflects the market's inherent unpredictability and the impact of unforeseen events.

For an ARMA model, the conditional mean μ_t is expressed as:

$$\mu_t = \phi_0 + \sum_{i=1}^{k} \beta_i x_{it} + \sum_{i=1}^{p} \phi_i r_{t-i} - \sum_{j=1}^{q} \theta_j a_{t-j}, \tag{3.2.3}$$

where ϕ_0 is a constant, ϕ_i and θ_j are coefficients for past values of r_t and a_t respectively, and β_i are coefficients for external variables x_{it}. These external variables are included in the model to explain variations in the conditional mean that cannot be captured by the time series' past values.

Combining Eq. 3.2.2 and Eq. 3.2.3, the conditional variance of r_t given the past information F_{t-1} can be defined as:

$$\sigma_t^2 = \text{Var}(r_t|F_{t-1}) = \text{Var}(a_t|F_{t-1}). \tag{3.2.4}$$

Volatility models focus on the evolution of the conditional variance σ_t^2, using the ARMA residuals a_t. These models fall into two broad categories: the first uses a precise function to dictate the progression of σ_t^2, the second employs a random

equation; we focus on the first.

3.2.2 The ARCH effect

To model a time series' volatility, typically, models assume conditional heteroscedasticity of the ARMA residuals ($a_t = r_t - \mu_t$), signifying the non-static variability of a series over time. This assumption allows for the modelling of volatility clustering, as even if the original residuals are uncorrelated, there will be significant autocorrelation in the squared residuals—as shown by Tsay (2005).

The presence of volatility clustering within ARMA residuals is called the autoregressive conditional heteroscedastic (ARCH) effect. If the residuals exhibit the ARCH effect, it is possible to utilise an ARCH model to quantify the conditional volatility by explicitly modelling the variance as a function of past squared residuals. The ARCH effect can be identified using various tests, including the Lagrange Multiplier test (Breusch and Pagan, 1979) and Engle's ARCH test (Engle, 1982).

3.2.3 The ARCH model

The autoregressive conditional heteroskedasticity (ARCH) model, pioneered by Engle (1982) relies on two key observations: the shock a_t in the return series is serially uncorrelated yet still has dependencies; the dependency in a_t can be articulated using a quadratic function based on its lagged values. Using these observations, a_t can be considered as a combination of a predictable component of the series σ_t and some random element, ε_t, left after modelling the predictable component. That is, $a_t = \sigma_t \varepsilon_t$, where

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \cdots + \alpha_m a_{t-m}^2, \tag{3.2.5}$$

given $\{\sigma_t\}$ represents a series of independent and identically distributed (i.i.d) random variables with mean of zero and variance of one; $\sigma_0 > 0$ and $\sigma_i \geqslant 0$ for $i > 0$;

and the distribution of ε_t is modelled as a standard Normal distribution, a Student's t-distribution, or a more generalised error distribution.

Large shocks in the past, denoted by $\{a_{t-i}^2\}$, elevate the present conditional variance, σ_t^2. Consequently, the residuals a_t exhibit large absolute values following significant shocks—so, a significant shock does not just signal a wider variance, it increases the likelihood of subsequent high-magnitude shocks, encapsulating volatility clustering.

ARCH models are limited by their inability to model longer-term volatility dependencies without significantly increasing the model's complexity. This increase in complexity can cause overfitting, making the model more sensitive to the idiosyncrasies of the training data and less capable of forecasting accurately.

3.2.4 The GARCH model

The generalized autoregressive conditional heteroskedasticity (GARCH) model introduced by Bollerslev (1986) builds on the ARCH model by incorporating past conditional variances. This captures longer memory in the volatility process without an extensively long lag structure.

For the GARCH model, σ_t^2 is modelled using the past squared errors a_{t-i}^2—as per the ARCH model—and also, the past conditional variances σ_{t-j}^2. Additionally, a base level of variance is set α_0. As such, the GARCH(m, s) process is defined as:

$$a_t = \sigma_t \varepsilon_t \quad \text{where,} \tag{3.2.6}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2, \tag{3.2.7}$$

where α_i and β_j are the coefficients of the lagged squared residual term and the lagged conditional variance term, respectively, which are parameters to be optimised.

GARCH forecasting

For a GARCH$(1,1)$ model, an l-step ahead forecast from origin h is given by:

$$\sigma_h^2(l) = \frac{\alpha_0(1-(\alpha_0+\beta_1)^{l-1})}{1-\alpha_1-\beta_1} + (\alpha_1+\beta_1)^{l-1}\sigma_h^2(1). \tag{3.2.8}$$

The first term captures the long-term variance. The second term captures the short-term shocks and volatilities. With a longer period l, the forecasted variance converges to its long-term value, provided that the shock effects decrease with time (commonly referred to as the stability condition). This can be formalised, by observing that if $\alpha_1+\beta_1<1$, then

$$\sigma_h^2(l) \to \frac{\alpha_0}{1-\alpha_1-\beta_1}, \text{ as } l\to\infty. \tag{3.2.9}$$

For a general GARCH(m,s) model, the l-step ahead forecast from origin h can be computed iteratively. The forecast of the conditional variance at each future step can be derived by:

$$\sigma_{h+k}^2(l) = \alpha_0 + \sum_{i=1}^{m}\alpha_i E[a_{h+k-i}^2(l)] + \sum_{j=1}^{s}\beta_j\sigma_{h+k-j}^2(l), \tag{3.2.10}$$

where $E[a_{h+k-i}^2(l)]$ is the expected value of the past squared errors at step $h+k-i$. This iterative process allows incorporating both past shocks and past variances, thus providing a comprehensive forecast of the future conditional variance. The long-term forecast converges to:

$$\sigma_\infty^2 = \frac{\alpha_0}{1-\sum_{i=1}^{m}\alpha_i - \sum_{j=1}^{s}\beta_j}, \tag{3.2.11}$$

assuming the stability condition $\sum_{i=1}^{m}\alpha_i + \sum_{j=1}^{s}\beta_j < 1$ holds. This long-term

variance represents the equilibrium level to which the forecasted conditional variance converges over time.

GARCH models are very effective for modelling the volatility in the ARMA residuals. By incorporating both long-term variances and short-term shocks, the conditional variance is modelled as a function not only of past squared errors (as in ARCH models) but also of past conditional variances. This feature enables GARCH models to account for more extensive historical information in modelling volatility clustering. However, GARCH models are limited by their inability to account for the asymmetries of market responses, or the leverage effect introduced earlier.

3.2.5 The TGARCH model

The Threshold GARCH (TGARCH) model extends the GARCH framework to account for the asymmetric impact of shocks on volatility commonly observed in financial markets, developed by Alexander (2008). The TGARCH model defines the conditional variance σ_t^2 at time t, as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 + \sum_{k=1}^{n} \gamma_k I_{t-k} a_{t-k}^2, \tag{3.2.12}$$

where γ_k captures the asymmetric effect of shocks; I_{t-k} is an indicator function, where $I_{t-k} = 1$ if $a_{t-k} < 0$ and $I_{t-k} = 0$ otherwise; and the terms common to the GARCH model are as defined in Eq. 3.2.7.

When a_{t-k} is negative (indicating a negative shock), then the term $\gamma_k a_{t-k}^2$ is added to the conditional variance. Conversely, when a_{t-k} is positive, the additional term is not added. This reflects the fact that negative shocks tend to increase volatility more than positive shocks of the same magnitude (Alexander, 2008).

3.3 ARMA-GARCH model

The ARMA-GARCH model refers to the joint process of using the ARMA model to model the mean of a time series, with the GARCH model applied to the ARMA residuals to model and forecast volatility (considering clustering and longer-term variances). One can also use an ARMA-TGARCH model, which utilises TGARCH to account for the leverage effect.

A core argument in this thesis is the need for financial models to separate risk quantifiable through probability from uncertainty, where outcomes are unknown and not amenable to probabilistic prediction. To understand whether an ARMA-GARCH forecast accounts for this uncertainty, we must understand the model order selection and parameter tuning processes.

3.3.1 Order choice and parameter tuning

In the ARMA-GARCH model, the optimal model orders ((p, q) for ARMA and (m, s) for GARCH) and model parameter values must be determined. When using ARMA-GARCH models, firstly, the optimal model orders for the ARMA model are chosen, and then the parameters are tuned. Using the ARMA residuals, the ARCH effect is tested for, and then the optimal GARCH model orders and parameters are tuned.

Typically, the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) is applied to different order combinations from which the optimal combination is chosen, balancing goodness of fit and simplicity. However, AIC/BIC struggle when the underlying data has high volatility and frequent structural breaks, as shown by Pesaran and Timmermann (2001). McQuarrie and Tsai (1999) identified that AIC/BIC also struggle when data exhibits complex dynamics and non-linear relationships, leading to overfitting.

ARMA-GARCH model parameters are often tuned using Maximum Likelihood Estimation (MLE). MLE relies on the assumption of normally distributed residuals, which is rarely true as residuals are often non-normal, heavy-tailed, and skewed. As such, MLE may result in poor parameter choice and the severe underestimation of financial risk measures, as shown in research by Cont (2001) and Rachev (2003).

When tuning ARMA-GARCH model parameters using MLE, the parameters are chosen so that—based on historical data—the proposed model specification best explains the observed data. This process, therefore, assumes that past patterns and relationships will persist into the future. Similarly, the use of AIC/BIC in model order selection assumes that the best model for future data prediction is one that has performed well on past data. Such frequentist methods are rooted in long-run probabilities based on experiments; while useful in certain scenarios, these methods do not effectively address the unique challenges of dynamic financial environments where the data set at hand is distinct and not one among many hypothetical scenarios. Mallikarjuna and Rao (2019) demonstrated that no single frequentist forecasting model uniformly applies to all markets. Under this frequentist framework, ARMA-GARCH models fall short in accounting for true uncertainty that is not quantifiable or predictable from historical data.

3.3.2 Proposed solution

In contrast to frequentist statistics, Bayesian statistics provides a more intuitive and flexible framework particularly suited to financial markets. Bayesian methods consider parameters and hypotheses as probability distributions with fixed data rather than deterministic values and outcomes. The Bayesian approach allows for the incorporation of prior knowledge and expert insights into the analysis, updating the hypothesis probabilities as new data becomes available (Fornacon-Wood et al., 2022).

By using a Bayesian evidence approach for model order selection and Bayesian inference for parameter estimation, the Bayesian models account for uncertainties in the model-building process.

Bayesian methods are particularly suited for modelling complex financial time series, as they provide more precise and robust models, accounting for the inherent uncertainties of financial markets—as demonstrated in research, for example, by Geweke (2001) and West and Harrison (2006). Adopting a Bayesian approach to ARMA-GARCH modelling will help us achieve our first objective to develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

Chapter 4

Bayesian Forecasting

Part of the reason for this research is the idea that financial models and investment strategies need to give greater consideration to tail risk and uncertainty—motivating the first thesis objective: to develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

In Chapter 2, fundamental characteristics of financial time series were established, like stationarity, which ensures the stability of statistical properties through time, and autocorrelation, which suggests past returns may indicate future returns. In Chapter 3, the ARMA-GARCH model for forecasting returns was introduced, which takes advantage of these properties, combining ARMA's proficiency in predicting the mean of a series with GARCH's ability to model and forecast changing volatility. The frequentist approach to model order selection and parameter tuning has several limitations due to its reliance solely on historical data. Whilst such an approach can handle quantifiable risks, it falls short in addressing true uncertainty relating to situations that are not quantifiable or predictable from historical data.

We propose developing a Bayesian approach to ARMA-GARCH models, which allows for the incorporation of prior knowledge and direct consideration of the uncertainty in financial data and models. Such an approach has the advantage of yielding

a probability distribution of forecasted returns rather than a point estimate (as in the frequentist framework).

This chapter develops the technical aspects of this application, demonstrating how Bayesian inference can enhance the predictive capabilities of ARMA and GARCH models, making them more responsive to the uncertainties of financial markets.

4.1 Bayesian inference

4.1.1 Introduction

This section briefly introduces Bayesian inference, which updates evolving probability distributions of parameters and hypotheses as new information becomes available. We start with key definitions, according to Gelman et al. (2013).

4.1.2 Key notation

To make probability statements on the distribution of some observed value y—which relates to the true unobservable data $\tilde{y}$—and its underlying parameters θ, a model can be established for the *joint probability distribution* of θ and y. This joint distribution is obtained as the product of two densities $p(\theta, y) = p(y|\theta)p(\theta)$, where $p(\theta)$ denotes the *prior distribution* of the parameters θ and $p(y|\theta)$ denotes the *sampling or likelihood distribution* of observing y given θ.

Bayes' rule can then condition this joint distribution on the observed value of the data y, to obtain the *posterior density*:

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)}, \tag{4.1.1}$$

where $p(y) = \sum_\theta p(y|\theta)p(\theta)$, or $p(y) = \int p(\theta)p(y|\theta)d\theta$ for the discrete and continuous

cases respectively.

The *prior predictive distribution*, which is the distribution of the unknown but observable y before the data is considered, is defined as:

$$p(y) = \int p(y, \theta) d\theta = \int p(\theta) p(y|\theta) d\theta. \tag{4.1.2}$$

After observing data y, the distribution of $\tilde{y}$ can be predicted with:

$$p(\tilde{y}|y) = \int p(\tilde{y}, \theta|y) d\theta \tag{4.1.3}$$

$$= \int p(\tilde{y}|\theta, y) p(\theta|y) d\theta \tag{4.1.4}$$

$$= \int p(\tilde{y}|\theta) p(\theta|y) d\theta. \tag{4.1.5}$$

This is called the *posterior predictive distribution* of $\tilde{y}$: posterior as it is conditional on the observed y and predictive as it is a prediction of an unobservable $\tilde{y}$ (Gelman et al., 2013).

4.1.3 Bayesian evidence

Bayesian model selection employs Bayesian evidence (or marginal likelihood) to determine which model best explains our observed data y. Simply put, Bayesian evidence is the probability of the observed data given a specific model, integrating over all possible parameter values θ—thereby considering any uncertainty within these parameters. For a model M, the Bayesian evidence is defined as

$$p(y|M) = \int p(y|\theta, M) p(\theta|M) d\theta, \tag{4.1.6}$$

where $p(y|\theta, M)$ is the likelihood function and $p(\theta|M)$ is the prior distribution of the parameters under M.

Model selection

Bayesian evidence in model comparison balances model fit (likelihood) with model complexity (prior), so a model that better explains the data without additional complexity will generally have higher Bayesian evidence. To compare two models, say M_1 and M_2, *Bayes factor* is utilised:

$$K = \frac{p(y|M_1)}{p(y|M_2)} \tag{4.1.7}$$

This ratio quantifies how much more likely the data is under model M_1 compared to M_2; that is, if $K > 1$, then M_1 is more strongly supported by the data, while $K < 1$ favours M_2. For multiple models, ranking them based on Bayesian evidence ascertains the model which best explains the observable data.

4.1.4 Bayesian inference conclusion

From the frequentist viewpoint, probability is a measure of the long-term frequency of events; in contrast, Bayesian statistics views probability as a measure of uncertainty. Bayesian inference allows for the inclusion of prior knowledge and the ability to update beliefs in light of new evidence. This is formalised through Bayes' Theorem, which updates the prior distribution $p(\theta)$ with the likelihood of the observed data $p(y|\theta)$ to arrive at the posterior distribution $p(\theta|y)$, capturing the updated beliefs about the uncertain parameter θ after observing data y.

The Bayesian framework recognises that uncertainty is not merely about the variability in repeated sampling but encompasses the lack of knowledge about a single event or parameter. For a Bayesian statistician, the probability of an event is a statement about an individual's degree of belief based on their own intuition and available information, including prior experiences and current evidence. This

contrasts with the frequentist approach, where probabilities are considered objective and solely based on long-term frequencies under repeated sampling conditions (Gelman et al., 2013).

The Bayesian approach to uncertainty is encapsulated in the posterior predictive distribution $p(\tilde{y}|y)$, which is the prediction for unobservable $\tilde{y}$ based on the observed data y. This predictive distribution reflects all the uncertainty about the model parameters and about future observations, integrating out the parameters using their posterior distributions.

By embracing the principles of Bayesian probability, one can make more informed decisions under uncertainty, reflecting both the known and the unknown in a cohesive and mathematically rigorous manner—explored in the general context by Gelman et al. (2013). Such an approach clearly has value in the ARMA-GARCH modelling framework.

4.2 Bayesian ARMA-GARCH model

Bayesian inference for ARMA and GARCH parameter tuning integrates prior knowledge and expert insights into the estimation process, treating model parameters not as fixed entities but as variables with their own probability distributions. Such a probabilistic approach allows for a more thorough exploration of the parameter space, ultimately providing a probability distribution of forecasted returns rather than a single-point estimate.

4.2.1 Bayesian ARMA model

ARMA model

For a time series r_t, an ARMA(p, q) model is defined as:

$$r_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + a_t + \sum_{j=1}^{q} \theta_j a_{t-j}, \tag{4.2.1}$$

where r_t is the time series value at time t, ϕ_0 is a constant, ϕ_i are AR coefficients, a_t is the error term, and θ_j are MA coefficients.

1. Prior distributions

In Bayesian ARMA modelling, the selection of prior distributions for the model parameters is crucial. For the ARMA parameters, weakly informative priors can be used to incorporate some prior knowledge while allowing the data to play a dominant role in the inference. A common choice for weakly informative priors on the AR and MA coefficients is the Normal distribution.

Let $\pi(\phi_i)$ and $\pi(\theta_j)$ denote the prior distributions for the AR and MA coefficients, respectively. A typical choice is the Normal prior distribution defined by:

$$\pi(\phi_i) \sim \mathcal{N}(0, 0.1), \tag{4.2.2}$$

$$\pi(\theta_j) \sim \mathcal{N}(0, 0.1). \tag{4.2.3}$$

This choice of prior is weakly informative and crucially, it keeps the parameter values close to zero, thereby reducing the risk of the characteristic roots lying outside the unit circle, which is necessary for stationarity and invertibility in ARMA processes.

Alternative prior variances can be used, depending on the specific context. Sensitivity analysis should be conducted on varying variance values to assess the ro-

bustness of the model results to different prior specifications.

2. Likelihood Function

The likelihood of observing the return data r given the chosen parameters is reflected in the likelihood function:

$$\mathcal{L}(r|\phi,\theta) = \prod_{t=1}^{T} f(r_t|\phi,\theta,r_{t-1},\ldots,r_{t-p},a_{t-1},\ldots,a_{t-q}), \tag{4.2.4}$$

where f is the probability density function of r_t given the parameters and past values. The construction of the likelihood function should consider the properties of financial returns, such as fat tails, volatility clustering, and potential non-linear dependencies.

The daily log returns for our core data set have a non-Normal distribution with heavy tails, negative skewness, and positive excess kurtosis. Hence, a skewed Student's t distribution is likely most appropriate. The probability density function f can be specified as:

$$f(r_t|\nu,\mu,\sigma,\alpha) \sim \text{Skewed-}t(\nu,\mu,\sigma,\alpha), \tag{4.2.5}$$

where ν is the degrees of freedom, μ is the location parameter, σ is the scale parameter, and α is the skewness parameter. The skewed Student's t-distribution is defined by its probability density function (PDF) as follows:

$$f(x|\nu,\mu,\sigma,\alpha) = \begin{cases} \frac{bc}{\sigma\sqrt{\nu}}\left(1+\frac{1}{\nu}\left(\frac{b(x-\mu)}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}, & \text{if } x<\mu, \\ \frac{bc}{\sigma\sqrt{\nu}}\left(1+\frac{1}{\nu}\left(\frac{b(x-\mu)}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}, & \text{if } x\geqslant\mu, \end{cases} \tag{4.2.6}$$

where

$$b = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi(\nu-2)}},$$
$$c = \left(1 + \left(\frac{\alpha}{\sqrt{\nu}}\right)^2\right)^{-\frac{\nu+1}{2}}.$$

This approach allows for more flexibility in modelling the distribution of returns, thereby providing a more accurate representation of the underlying data-generating process.

3. Posterior distribution

The posterior distribution combines our prior beliefs with the observed data, updating our understanding of the parameter values. This posterior distribution is proportional to the product of the likelihood and the priors:

$$\pi(\phi, \theta|r) \propto \mathcal{L}(r|\phi, \theta)\pi(\phi)\pi(\theta). \tag{4.2.7}$$

Due to the complexity of these models, numerical methods like Markov Chain Monte Carlo (MCMC) are often employed to approximate the posterior. The choice of these methods and their tuning parameters (like the number of iterations or burn-in period) should be made carefully, as the choice can significantly impact the accuracy and computational efficiency of the inference process.

From this posterior distribution, it is possible to obtain the predictive posterior distribution of future asset returns, given by:

$$p(\tilde{r}|r) = \int \mathcal{L}(\tilde{r}|\phi, \theta)\pi(\phi, \theta|r)d\phi d\theta. \tag{4.2.8}$$

Implementation testing

To ensure the reliability of the Bayesian ARMA model, it is crucial to validate the model by checking its ability to predict out-of-sample data accurately. One way to do this is to use posterior predictive checks, where the simulated returns data is compared to the actual observed data. If there are discrepancies, this could indicate a model misspecification or insufficiently informative priors. Additionally, sensitivity analysis can be performed to assess how sensitive the results are to different prior choices. This analysis helps to guide the refinement of prior distributions.

4.2.2 Bayesian GARCH model

The Bayesian ARMA model uses a probabilistic framework to capture the linear relationships in time series data. However, it is essential to model the volatility in financial data using GARCH models on ARMA residuals.

GARCH model

The GARCH process is applied to ARMA model residuals to model volatility, defined as:

$$a_t = \sigma_t \varepsilon_t \text{ where,} \tag{4.2.9}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i a_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2, \tag{4.2.10}$$

where ε_t is an independent and identically distributed (i.i.d.) random variable with zero mean and unit variance. The coefficients α_i and β_j are GARCH model parameters, which must meet the positivity condition:

$$\alpha_0 > 0, \quad \alpha_i \geqslant 0 \quad \text{for all } i, \quad \beta_j \geqslant 0 \quad \text{for all } j. \tag{4.2.11}$$

This condition ensures that all coefficients in the GARCH equation contribute non-negatively to the conditional variance, maintaining its positivity. Additionally, the GARCH parameters must satisfy the following summability condition:

$$\sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{s} \beta_j < 1. \tag{4.2.12}$$

This condition ensures that the model is stationary, guaranteeing that the series has a finite unconditional variance and that the volatility process reverts to a long-term mean.

1. Prior distributions

The GARCH prior distributions can be modelled using a Gamma distribution, to ensure that the GARCH parameters are strictly positive, thereby ensuring that the GARCH model is positive (as required). Other options like Inverse-Gamma, Beta, or Log-Normal distributions might be more appropriate depending on the characteristics of the financial returns in the training data. The chosen prior should accommodate the expected range and volatility behaviour observed historically. For instance, if volatility exhibits long-range dependence or persistence, priors should be chosen to reflect such features.

Let $\pi(\alpha_i)$ and $\pi(\beta_j)$ denote the prior distributions for the GARCH coefficients, respectively. A Gamma distributions can be defined by:

$$\pi(\alpha_i) \sim \text{Gamma}(\kappa_\alpha, \lambda_\alpha), \tag{4.2.13}$$

$$\pi(\beta_j) \sim \text{Gamma}(\kappa_\beta, \lambda_\beta), \tag{4.2.14}$$

where κ is the shape hyperparameter and λ is the scale hyperparameter, tuned based on previous empirical studies or expert judgement.

2. Likelihood Function

In the context of a GARCH model, where a_t represents the errors or ARMA residuals being modelled, the likelihood function is essential for quantifying the probability of the observed residuals given a set of model parameters. The likelihood function is the product of the probability densities of each observed residual, conditioned on past values and the conditional variance estimated by the GARCH parameters:

$$\mathcal{L}(a|\alpha,\beta) = \prod_{t=1}^{T} f(a_t|\alpha,\beta,a_{t-1},\ldots,a_{t-m},\sigma_t^2,\ldots,\sigma_{t-s}^2). \tag{4.2.15}$$

Here, f represents the probability density function (pdf) of the residuals a_t, given the GARCH parameters α and β, and conditioned on the previous residuals $a_{t-1},\ldots,a_{t-m}$ and past conditional variances $\sigma_t^2,\ldots,\sigma_{t-s}^2$. The pdf f characterizes the distribution of the residuals, often assumed to follow a Normal distribution in basic GARCH models but potentially following other distributions such as Student's t-distribution in more complex models. This function is formulated based on the conditional volatility modelled by the GARCH process.

3. Posterior distribution

The posterior distribution is the updated belief about the model's parameters after observing the market data, derived by combining the prior distribution of the GARCH parameters with the likelihood of the observed data under these parameters:

$$\pi(\alpha,\beta|a) \propto \mathcal{L}(a|\alpha,\beta)\pi(\alpha)\pi(\beta). \tag{4.2.16}$$

This distribution is often approximated or sampled using numerical methods, such as Markov Chain Monte Carlo (MCMC), due to the complexity of obtaining an analytical solution. The posterior distribution is used to make probabilistic statements

about future volatility and assess the model's predictive performance.

From this posterior distribution, it is possible to obtain the predictive posterior distribution of future asset volatility, or errors, given by:

$$p(\tilde{a}|a) = \int (\tilde{a}|\alpha, \beta)\pi(\alpha, \beta|a)d\alpha d\beta. \tag{4.2.17}$$

Bayesian TGARCH

The Bayesian framework can be adapted for TGARCH modelling of the ARMA residuals to allow for different responses in volatility to positive and negative shocks. Bayesian TGARCH modelling of the ARMA residuals is an effective method for capturing volatility in financial time series. This method integrates prior knowledge, updating parameter estimates with observed data, and provides greater insight into financial forecasting incorporating uncertainties.

For the TGARCH model, an additional prior distribution is required for the parameter γ_k, which should be chosen carefully to reflect the asymmetry in volatility—a Gamma distribution can still be used:

$$\pi(\gamma_k) \sim \text{Gamma}(\kappa_\gamma, \lambda_\gamma). \tag{4.2.18}$$

The likelihood function for TGARCH differs from that of GARCH due to the additional term γ_k, to reflect the probability of observed returns given the asymmetric impact on volatility:

$$\mathcal{L}(a|\alpha, \beta, \gamma) = \prod_{t=1}^{T} f(a_t|\alpha, \beta, \gamma, a_{t-1}, \ldots, a_{t-m}, \sigma^2_{t-1}, \ldots, \sigma^2_{t-s}, I_{t-1}, \ldots, I_{t-n}). \tag{4.2.19}$$

Lastly, the posterior distribution in the TGARCH context incorporates the likeli-

hood function considering the asymmetric effects:

$$\pi(\alpha, \beta, \gamma|a) \propto \mathcal{L}(a|\alpha, \beta, \gamma)\pi(\alpha)\pi(\beta)\pi(\gamma). \tag{4.2.20}$$

Implementation testing

Validating a Bayesian (T)GARCH model is crucial to ensure that it effectively captures volatility dynamics. This process usually involves analysing the model's residuals to check for any remaining patterns or autocorrelations, which could suggest the model is not a good fit. Posterior predictive checks are also useful in this case, as they can reveal how well the model captures the distributional properties of financial returns, such as the heavy tails and clustering of volatility.

4.3 Bayesian ARMA-GARCH order choice

For the ARMA(p, q) model, autocorrelation and partial autocorrelation analysis provide initial estimates for the required AR and MA terms, p and q. For each combination, the Bayesian evidence can then be calculated—quantifying how well each model explains the observed data—by integrating the product of the likelihood and the prior over all possible parameter values:

$$P(r|M) = \int \mathcal{L}(r|\phi, \theta, M)\ \pi(\phi|M)\ \pi(\theta|M)\ d\phi\ d\theta. \tag{4.3.1}$$

Similarly, to determine the orders m and s in a GARCH(m, s) model, one can establish GARCH order candidates by analysing the ARMA residuals through statistical tests for ARCH effects, and testing the significance of lagged terms. Then, for each combination of potential orders, the Bayesian evidence can be calculated:

$$P(r|M) = \int \mathcal{L}(a|\alpha, \beta, M)\ \pi(\alpha|M)\ \pi(\beta|M)\ d\alpha\ d\beta. \tag{4.3.2}$$

4.3.1 Bayesian evidence computation

The computation of Bayesian evidence can be challenging, especially in financial time series forecasting models, as the likelihood functions are complex and high-dimensional, making the computation analytically intractable. Although numerical methods like Markov Chain Monte Carlo (MCMC) can be used, their associated computational cost is often prohibitively expensive. To address this, many advanced methods have been developed.

One approach is to use vertical likelihood representations, which transform the multi-dimensional integrals into equivalent one-dimensional integrals. This simplifies the computation process, making it more efficient and tractable, especially in high-dimensional parameter spaces. Vertical likelihood representations could be advantageous for Bayesian evidence estimation in ARMA-GARCH models due to their complex, high-dimensional parameter spaces.

The multicanonical Monte Carlo method (MMC) is an example of a vertical likelihood representation method, but to date, it has not been applied for the purpose of Bayesian evidence estimation. This method is known for its efficiency in sampling from complex, multimodal distributions, so its application could lead to more accurate and efficient Bayesian evidence calculation.

Llorente et al. (2023) examine various computational methods for Bayesian evidence estimation, although their work focuses on Bayesian evidence estimation in statistical physics applications, it provides valuable insights into the theoretical and practical aspects of Bayesian evidence computation. We draw on this work within the following sections.

The next sections introduce vertical likelihood representations and alternative non-parametric approaches, including the original presentation of MMC in this context. Up to now, no comprehensive comparison of Bayesian evidence methods for

financial time series applications has been published. In this review, a comparison and analysis, with numerical examples, is presented. This review is distinct from that of Llorente et al. (2023) as it specifically addresses the application to financial forecasting models and directly compares vertical likelihood representations to alternate non-parametric approaches.

4.3.2 Vertical likelihood representations

Most traditional approaches to determining integrals rely on integrating over the whole parameter space—more or less uniformly exploring it. By looking across the parameter space, such methods are considered *horizonal* methods. In contrast, one can transform the calculation of the marginal likelihood function into a more easily approximated form by converting the multidimensional integral over the parameter space into a one-dimensional integral over a single variable; focusing sampling on those areas which are of primary interest. Such methods can be considered *vertical likelihood representations*.

One example is nested sampling—detailed below and in the paper by Llorente et al. (2023). Another method is the multicanonical Monte Carlo method (MMC), detailed later in Chapter 8. To date, MMC has not been employed for the purpose of Bayesian evidence estimation; we will explore its usefulness in this context shortly.

Nested Sampling

Nested sampling was specifically developed for estimating Bayesian evidence, as detailed in the work by Skilling (2006). It transforms the high-dimensional integral required for Bayesian evidence into a one-dimensional integral by integrating over the 'prior mass' rather than directly over the entire parameter space. Through iterative sampling, the parameter space is systematically narrowed, focusing samples in regions of higher likelihood by replacing the lowest-likelihood sample with a new

one from the prior distribution with a higher likelihood.

A set of samples is generated from the prior distribution, from which points are iteratively refined based on the likelihood function. The Bayesian evidence $Z = p(y|M)$ can be estimated as:

$$Z = \int_0^1 \mathcal{L}(\theta)\, dX, \tag{4.3.3}$$

where $\mathcal{L}(\theta)$ denotes the likelihood function evaluated at a parameter set θ, and X represents the prior mass. The integral is estimated by summing over a series of likelihoods, each weighted by a differential amount of prior mass.

Skilling (2006) demonstrates that the algorithm's evolution is guided by the shape of the likelihood contours, which remain invariant under monotonic re-labelling. This invariance implies that the path of convergence is independent of the actual likelihood values, focusing solely on the prior mass reduction, thus facilitating convergence. The convergence of nested sampling is theoretically guaranteed as the number of iterations tends to infinity, provided that the algorithm adequately explores the entire parameter space. The accuracy of the evidence estimate improves with the number of live points (N) and iterations, with uncertainties decreasing proportionally to $1/\sqrt{N}$ (Skilling, 2006). However, practical convergence depends on ensuring that sufficient iterations are performed to capture the bulk of the posterior mass, typically indicated by a plateau in the accumulation of the evidence estimate. This convergence criterion is often a matter of judgment, with a recommendation to continue iterations until the evidence estimate stabilises or an analytical upper bound on the likelihood is reached.

Nested sampling offers a robust and efficient approach for estimating Bayesian evidence in the context of model selection. We reserve further discussion of the

advantages, limitations, and applicability to Bayesian ARMA-GARCH models for the later comparison in Section 4.4.

Algorithm 1 Nested Sampling Algorithm for Bayesian Evidence

Input: Number of live points N, model M, data y
Initialise a set of live points $\{\theta_i\}_{i=1}^{N}$ from the prior $p(\theta|M)$
Evaluate the likelihood $\mathcal{L}_i = p(y|\theta_i, M)$ for each live point
Set initial evidence estimate $Z = 0$ and prior mass $X_0 = 1$
for $i = 1$ to N **do**
    Find the live point with the lowest likelihood, $\mathcal{L}_{\min}$
    Calculate the weight for the current iteration $w_i = X_{i-1} - X_i$, where $X_i = e^{-i/N}$
    Update evidence estimate: $Z \leftarrow Z + \mathcal{L}_{\min} \times w_i$
    Remove the live point with likelihood $\mathcal{L}_{\min}$
    Replace it with a new sample from the prior with likelihood $\mathcal{L} > \mathcal{L}_{\min}$
end for
Increment Z by $\frac{1}{N}\left(\sum_{i=1}^{N} \mathcal{L}_i\right) \times X_N$
Return: Bayesian evidence estimate Z

Multicanonical Monte Carlo

In Chapter 8 we extensively detail the multicanonical Monte Carlo (MMC) method. Here, we provide enough detail to understand MMC in the context of Bayesian evidence estimation. MMC is a form of importance sampling which splits the state space of a performance variable into a set of small bins. Then MMC iteratively constructs a so-called flat-histogram importance sampling distribution that can assign equal probabilities to each bin, to ensure sampling occurs across the target distribution. MMC can explore complex, high-dimensional spaces by accurately re-

constructing the performance variable's entire distribution, making it suitable for Bayesian evidence computation.

Starting from an initial distribution $p(x)$, MMC iteratively adjusts its sampling distribution to drive sampling towards the regions of low probability. Specifically, given an initial importance sampling distribution $q_0(x)$ and a set of parameters Θ_0, MMC iteratively refines an importance sampling distribution, given by:

$$q_t(x) \propto p(x) \cdot \Theta_t(x), \tag{4.3.4}$$

where t denotes the iteration step, and Θ_t represents a set of weights adjusted at each step to flatten the histogram of sampled values.

In the context of Bayesian model selection, MMC facilitates the computation of the Bayesian evidence through sampling from a sequence of modified distributions, which eventually leads to efficient sampling from the posterior distribution $p(\theta|y, M)$. The evidence $p(y|M)$ can then be estimated using these samples, typically involving techniques such as importance sampling or histogram-based methods.

This method has been shown to be asymptotically exact, meaning that as the number of iterations increases, the estimated probability distribution converges to the true distribution. The key to this exactness lies in the iterative reweighting procedure that adjusts the sampling distribution until it closely approximates the desired uniform distribution across the entire state space (Berg and Neuhaus, 1992).

MMC offers a new, powerful approach to Bayesian evidence estimation. Its ability to navigate through complex parameter spaces and adaptively sample from posterior distributions makes it a valuable technique in this context. The exactness of the Bayesain evidence estimation largely depends on how well the algorithm has obtained a flat-histogram importance sampling distribution. We reserve further dis-

cussion of the advantages, limitations and applicability to Bayesian ARMA-GARCH models for the comparison in Section 4.4.

Algorithm 2 Multicanonical Monte Carlo for Bayesian evidence

1: **Input**: Initial distribution $q_0(x)$, initial parameters Θ_0, target distribution $p(x)$
2: **for** each iteration t **do**
3: Modify the distribution: $q_t(x) \propto p(x) \cdot \Theta_t(x)$
4: Sample from $q_t(x)$ and adjust Θ_t to flatten histogram
5: **end for**
6: Estimate Bayesian evidence using histogram method
7: **Return**: Bayesian evidence estimate $p(y|M)$

4.3.3 Alternative non-parametric approaches

Following the discussion on vertical likelihood representations, we introduce three key non-parametric approaches for Bayesian evidence estimation: bridge sampling, thermodynamic integration and steppingstone sampling.

Bridge Sampling

Bridge sampling is an effective method for estimating the Bayesian evidence (or marginal likelihood) by relating two different probability distributions. The Bayesian evidence is critical for model comparison. The bridge sampling identity, as defined by Meng and Wong (1996), estimates the ratio of the evidences of two models M_1 and M_2 with respective posterior distributions $p(\theta|y, M_1)$ and $p(\theta|y, M_2)$. The identity is given by:

$$\frac{p(y|M_1)}{p(y|M_2)} = \frac{\mathbb{E}_{p(\theta|y,M_2)}\left[\frac{p(y|\theta,M_1)p(\theta|M_1)}{q(\theta)}\right]}{\mathbb{E}_{p(\theta|y,M_1)}\left[\frac{p(y|\theta,M_2)p(\theta|M_2)}{q(\theta)}\right]}, \tag{4.3.5}$$

where $q(\theta)$ is a suitably chosen proposal distribution, and $\mathbb{E}_{p(\theta|y,M_i)}$ denotes the

expectation under the posterior distribution of model M_i.

The proposal distribution $q(\theta)$ serves as a bridge between the posterior distributions of the models being compared. It should ensure sufficient overlap between the posterior distributions $p(\theta|y, M_1)$ and $p(\theta|y, M_2)$. If the proposal distribution does not sufficiently cover the high probability region of the posterior, the estimator can suffer from high variance and potential bias, leading to inaccurate evidence estimation (Gronau et al., 2020). Common choices for $q(\theta)$ include:

- **Mixture distributions**: Combines the posterior distributions of the models being compared. Provides good overlap and can balance accuracy and computational efficiency.
- **Adaptive distributions**: Tailored to specific properties of the posterior distributions involved. More complex but can offer better performance in certain scenarios.

In their work, Meng and Wong (1996) demonstrated that an asymptotically optimal bridge function can minimise the relative mean-square error of the estimator. Under such a condition, the evidence estimate converges to the true value. This optimality is achieved by balancing the contributions from both the target and the proposal distributions, ensuring that the bridge function efficiently bridges the gap between them.

Bridge sampling is a valuable tool in Bayesian model selection. We will discuss the advantages, limitations, and applicability to Bayesian ARMA-GARCH models in the later comparison in Section 4.4.

Algorithm 3 Bridge Sampling for Bayesian Evidence Estimation

1: **Input**: Posterior distributions $p(\theta|y, M_1)$ and $p(\theta|y, M_2)$, proposal distribution $q(\theta)$
2: Obtain samples $\{\theta_i^{(1)}\}_{i=1}^{N_1}$ from posterior $p(\theta|y, M_1)$ and $\{\theta_i^{(2)}\}_{i=1}^{N_2}$ from posterior $p(\theta|y, M_2)$
3: Choose a proposal distribution $q(\theta)$
4: **for** each sample $\theta_i^{(1)}$ from $p(\theta|y, M_1)$ **do**
5: Compute weight $w_i^{(1)} = \frac{p(y|\theta_i^{(1)}, M_2)p(\theta_i^{(1)}|M_2)}{q(\theta_i^{(1)})}$
6: **end for**
7: **for** each sample $\theta_i^{(2)}$ from $p(\theta|y, M_2)$ **do**
8: Compute weight $w_i^{(2)} = \frac{p(y|\theta_i^{(2)}, M_1)p(\theta_i^{(2)}|M_1)}{q(\theta_i^{(2)})}$
9: **end for**
10: Estimate Bayes factor $K = \frac{\frac{1}{N_2}\sum_{i=1}^{N_2} w_i^{(2)}}{\frac{1}{N_1}\sum_{i=1}^{N_1} w_i^{(1)}}$
11: **Return**: Bayesian evidence ratio K

Thermodynamic Integration

Thermodynamic integration (TI) bridges between the prior and the posterior of a single model, as proposed by Swendsen and Wang (1986). The method is rooted in statistical mechanics, where the analogy between statistical thermodynamics and Bayesian inference is established.

TI introduces a temperature parameter β, which scales the likelihood function, creating a path between the prior distribution ($\beta = 0$) and the posterior distribution ($\beta = 1$). Formally, the Bayesian evidence $p(y|M)$ can be expressed in terms of the temperature integral (Lartillot and Philippe, 2006):

$$\ln p(y|M) = \int_0^1 \mathbb{E}_\beta[\ln p(y|\theta, M)]d\beta,$$

where $\mathbb{E}_\beta[\cdot]$ denotes the expectation with respect to the distribution

$$p_\beta(\theta|y, M) \propto p(y|\theta, M)^\beta p(\theta|M)^{1-\beta}.$$

The integral effectively averages the log-likelihood over a continuum of distributions between the prior and posterior. This algorithm integrates over the parameter β by discretising the interval $[0, 1]$ into K steps and approximating the integral using the trapezoidal rule. The sampled θ_i values are used to compute the expectation of the log-likelihood at each intermediate distribution, which is then averaged to estimate the log evidence.

Convergence of the TI algorithm to the true Bayesian evidence is generally assured under specific conditions, as outlined by Lartillot and Philippe (2006). First, the choice of the temperature path is essential. The parameter β, which scales the likelihood function, needs to vary smoothly from 0 to 1, ensuring a continuous bridge between the prior and posterior distributions–especially in regions where the likelihood changes rapidly (Knuth et al., 2015).

Secondly, the sample size at each β step must be sufficiently large to provide reliable estimates of the expected log-likelihood. Insufficient sampling can lead to inaccurate integration and, consequently, poor convergence (Annis et al., 2019).

Finally, the integration method used to compute the area under the curve formed by the log-likelihood values across different β values must be accurate. Techniques like the trapezoidal rule or Simpson's rule can be applied to achieve this Annis et al. (2019). Convergence is generally assured if these conditions are met. However, difficulties arise in cases of phase transitions or multimodal distributions, where the path between the prior and posterior is not smooth.

Algorithm 4 Thermodynamic Integration for Bayesian Evidence

1: **Input**: Number of intermediate steps K, number of samples N, model M, data y

2: Define a sequence $\{\beta_k\}$ where $\beta_0 = 0$ and $\beta_K = 1$

3: **for** $k = 0$ to K **do**

4: Sample $\{\theta_i\}_{i=1}^N$ from the intermediate distribution $p_\beta(\theta|y, M) \propto p(y|\theta, M)^{\beta_k} p(\theta|M)^{1-\beta_k}$

5: Compute sample average $\overline{L}_k = \frac{1}{N} \sum_{i=1}^N \ln p(y|\theta_i, M)$

6: **end for**

7: Estimate Bayesian evidence:

$$\ln p(y|M) \approx \sum_{k=0}^{K-1} \frac{\beta_{k+1} - \beta_k}{2} (\overline{L}_k + \overline{L}_{k+1})$$

8: **Return**: Bayesian evidence estimate $\mathrm{e}^{(\ln p(y|M))}$

Steppingstone Sampling

The large number of integrals to be calculated during thermodynamic integration can have a significant computational cost. First introduced by Xie et al. (2010), steppingstone sampling (SSS) is a refinement of thermodynamic integration, offering a more efficient approach for estimating Bayesian evidence.

Steppingstone sampling involves a series of intermediate distributions that bridge the gap between the prior and posterior, similar to thermodynamic integration. However, SSS uses a discrete set of 'stepping stones' instead of a continuous path, making the computation more manageable.

The Bayesian evidence under SSS is estimated as a product of ratios of expectations between adjacent stepping stones. By defining a sequence of distributions parameterized by β_k, where $\beta_0 = 0$ is used for the prior and $\beta_K = 1$ for the posterior,

the Bayesian evidence can be estimated as:

$$\ln p(y|M) \approx \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \left(\ln \left(\frac{1}{N} \sum_{i=1}^{N} p(y|\theta_i, M)^{\beta_k - \beta_{k-1}} \right) \right), \tag{4.3.6}$$

where the expectation $\mathbb{E}_{\beta_k}[\cdot]$ denotes the expectation with respect to the power posterior distribution $p(\theta|y, M)^{\beta_k} p(\theta|M)^{1-\beta_k}$.

Instead of integrating over the entire path from the prior to the posterior (as is done in thermodynamic integration), SSS calculates the ratios of normalizing constants between successive pairs of power posteriors. By multiplying these ratios, the Bayesian evidence is obtained. By using a discrete set of distributions, SSS is more computationally efficient than continuous-path methods like thermodynamic integration.

The convergence of steppingstone sampling to the true Bayesian evidence is theoretically assured under certain conditions. For instance, the algorithm's efficacy is contingent on the proper selection of the sequence of intermediate distributions. The use of a Beta distribution for determining the temperature schedule, as highlighted by Xie et al. (2010), optimises the placement of these intermediate steps, ensuring a more stable and accurate convergence.

Empirical studies have shown that with an adequately large number of intermediate distributions, the steppingstone sampling provides a reliable estimate of the Bayesian evidence, reducing bias as the number of stepping stones increases (Annis et al., 2019). Conversely, improper implementation or insufficient stepping stones can impede its convergence and accuracy.

Algorithm 5 Steppingstone Sampling for Bayesian Evidence

1: **Input**: Number of intermediate distributions K, number of samples N, prior $p(\theta|M)$, posterior $p(\theta|y, M)$

2: Define a sequence of distributions $\{p_{\beta_k}(\theta|y, M)\}$ where $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$

3: **for** $k = 1$ to K **do**

4: Sample $\{\theta_i\}_{i=1}^N$ from the power posterior $p_{\beta_k}(\theta|y, M)$

5: Compute the ratio of normalizing constants:

$$r_k = \frac{1}{N} \sum_{i=1}^{N} p(y|\theta_i, M)^{\beta_k - \beta_{k-1}}$$

6: **end for**

7: Estimate Bayesian evidence:

$$\ln p(y|M) \approx \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \ln r_k$$

8: **Return**: Bayesian evidence estimate $\exp(\ln p(y|M))$

4.4 Bayesian evidence method comparison

This section examines the effectiveness of vertical likelihood representations (such as nested sampling and multicanonical Monte Carlo) and non-parametric alternatives (including bridge sampling, thermodynamic integration, and steppingstone sampling) in Bayesian evidence estimation by considering:

1. **Accuracy:** To measure how closely each method's output aligns with a true baseline solution; crucial for ensuring the selected model reflects the underlying data and its dynamics.

2. **Scalability:** To assess how well a method can adapt to increasing dimensionality; critical in the finance sector, where data volumes and complexity are constantly growing, a scalable solution is necessary to handle large datasets efficiently.

3. **Efficiency:** To measure the computational resources and time required for convergence; particularly important in time-sensitive financial applications.

4. **Complexity:** To consider complexity and interpretability. Simpler methods are often more desirable for practical implementation and understanding.

5. **Robustness:** To determine the sensitivity of each method to their parameters. A robust method performs consistently under parameter changes.

These criteria are applicable in the general setting, so we establish additional criteria specific to Bayesian evidence estimation for Bayesian ARMA-GARCH models:

1. **Efficient sampling.** The algorithm should employ efficient sampling methods to explore the entire parameter space of ARMA-GARCH models.

2. **Handling high-dimensional spaces:** The algorithm must be capable of operating effectively in high-dimensional spaces—like ARMA-GARCH models—avoiding the 'curse of dimensionality'.

3. **Tail events:** The algorithm should ensure the entire ARMA-GARCH parameter space is explored to capture tail events common in financial markets.

4.4.1 Numerical example

We introduce a demonstrative example to evaluate various aspects of each estimation method while having an easily obtainable baseline solution.

Prior distribution. Define an n-dimensional input vector $\mathbf{x}$ following a multivariate Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix $\mathbf{I}_n$ (an $n \times n$ identity matrix). The prior distribution for our example is given by:

$$\mathbf{x} \sim N(\mathbf{0}, \mathbf{I}_n). \tag{4.4.1}$$

Performance function and likelihood. Define a dependent variable $\tilde{y}$, as the sum of squares of the elements of $\mathbf{x}$, following a Chi-Squared distribution with n degrees of freedom:

$$\tilde{y} = \sum_{i=1}^{n} x_i^2, \quad \tilde{y} \sim \chi^2(n). \tag{4.4.2}$$

The observed noisy data y is modelled by the likelihood function:

$$p(y|\tilde{y}) = \exp\left(-\frac{(y-\tilde{y})^2}{2}\right), \tag{4.4.3}$$

quantifying the discrepancy between the observed y and true value $\tilde{y}$.

Posterior distribution. Applying Bayes' Theorem, the posterior distribution $p(\tilde{y}|y)$ is:

$$p(\tilde{y}|y) = \frac{p(y|\tilde{y})p(\tilde{y})}{p(y)}, \tag{4.4.4}$$

where $p(y)$ is the marginal likelihood or Bayesian evidence.

Bayesian evidence. The objective is to compute the Bayesian evidence $p(y)$ for varying degrees of freedom n and observation $y = 0$. Considered in terms of $\mathbf{x}$, the Bayesian evidence estimation is equivalent to calculating the expectation of the likelihood function, denoted by $f(\mathbf{x})$, given by:

$$\mathbb{E}[f(\mathbf{x})] = \int \exp\left(-\frac{(\sum_{i=1}^{n} x_i^2)^2}{2}\right) p(\mathbf{x}) d\mathbf{x}. \tag{4.4.5}$$

Numerical results

Each method's performance is assessed for varying degrees of freedom (DoF = $2, 5, 10, 18$). Numerical integration, specifically the `scipy.integrate.quad` function from Python's SciPy library, establishes a reference solution for our demonstrative example.

Thermodynamic integration and steppingstone sampling are implemented in two ways. The first is the proper implementation, with bridges adaptively chosen to optimally bridge between the prior and posteriors, denoted simply as "TI" and "SSS". To demonstrate the sensitivity to the bridging set-up, we also employ an evenly distributed bridging strategy denoted "TI-Equal" and "SSS-Equal', using equal intervals for bridging rather than an adaptive strategy.

The numerical results are given in Table 4.1 showing the mean and standard deviations (Std Dev) for each method and DoF—under 10 repeated experiments. The table also states the number of evaluations required for a viable and stable solution (# Evals). Table 4.2 shows the Mean Squared Errors (MSE) using a baseline solution. Table 4.3 shows the Coefficient of Variation = Standard Deviation / Mean.

	DoF 2		DoF 5		DoF 10		DoF 18	
Method	**# Evals**	**Mean (Std Dev)**	**# Evals**	**Mean (Std Dev)**	**# Evals**	**Mean (Std Dev)**	**# Evals**	**Mean (Std Dev)**
Baseline	-	4.38×10^{-1}	-	8.51×10^{-2}	-	1.79×10^{-3}	-	6.27×10^{-7}
NS	25000	4.55×10^{-1} (3.97×10^{-2})	121000	7.87×10^{-2} (5.97×10^{-3})	144000	1.86×10^{-3} (2.75×10^{-4})	225000	6.55×10^{-7} (1.74×10^{-7})
MMC	2000	4.19×10^{-1} (4.45×10^{-2})	4000	6.28×10^{-2} (5.92×10^{-3})	50000	1.67×10^{-3} (1.72×10^{-4})	1500000	5.89×10^{-7} (6.21×10^{-8})
BS	10000	4.34×10^{-1} (1.11×10^{-2})	10000	7.46×10^{-2} (6.55×10^{-3})	10000	1.94×10^{-3} (4.79×10^{-4})	10000	1.49×10^{-7} (3.04×10^{-7})
TI	10000	4.21×10^{-1} (1.65×10^{-2})	10000	8.21×10^{-2} (4.52×10^{-3})	10000	1.59×10^{-3} (1.49×10^{-4})	10000	5.95×10^{-7} (7.52×10^{-8})
TI-Equal	10000	3.71×10^{-1} (3.47×10^{-3})	10000	5.102×10^{-2} (1.57×10^{-3})	10000	1.40×10^{-3} (1.21×10^{-4})	10000	5.79×10^{-7} (4.58×10^{-8})
SSS	10000	4.16×10^{-1} (1.78×10^{-2})	10000	7.98×10^{-2} (3.26×10^{-3})	10000	1.64×10^{-3} (1.54×10^{-4})	10000	6.11×10^{-7} (6.58×10^{-8})
SSS-Equal	10000	2.53×10^{-1} (1.93×10^{-3})	10000	7.14×10^{-2} (1.75×10^{-3})	10000	1.59×10^{-3} (1.53×10^{-4})	10000	1.11×10^{-7} (1.08×10^{-8})

Table 4.1: Comparative results of Bayesian evidence methods

Method	DoF 2	DoF 5	DoF 10	DoF 18
NS	1.88×10^{-3}	7.69×10^{-5}	8.13×10^{-8}	3.08×10^{-14}
MMC	2.36×10^{-3}	5.32×10^{-4}	4.29×10^{-8}	5.27×10^{-15}
BS	1.38×10^{-4}	1.54×10^{-4}	2.51×10^{-7}	2.30×10^{-13}
TI	5.51×10^{-4}	2.93×10^{-5}	5.86×10^{-8}	6.68×10^{-15}
TI-Equal	4.54×10^{-3}	1.16×10^{-3}	1.66×10^{-7}	4.39×10^{-15}
SSS	8.20×10^{-4}	3.89×10^{-5}	4.54×10^{-8}	4.60×10^{-15}
SSS-Equal	3.42×10^{-2}	1.90×10^{-4}	6.30×10^{-8}	2.66×10^{-13}

Table 4.2: Mean squared error (MSE) of Bayesian evidence methods

Method	DoF 2	DoF 5	DoF 10	DoF 18
NS	8.72×10^{-2}	7.58×10^{-2}	1.48×10^{-1}	2.65×10^{-1}
MMC	1.06×10^{-1}	9.42×10^{-2}	1.03×10^{-1}	1.05×10^{-1}
BS	2.55×10^{-2}	8.78×10^{-2}	2.47×10^{-1}	2.05×10^{-1}
TI	3.91×10^{-2}	5.50×10^{-2}	9.31×10^{-2}	1.26×10^{-1}
TI-Equal	9.36×10^{-3}	3.08×10^{-2}	8.61×10^{-2}	7.90×10^{-2}
SSS	4.29×10^{-2}	4.08×10^{-2}	9.40×10^{-2}	1.08×10^{-1}
SSS-Equal	7.61×10^{-3}	2.45×10^{-2}	9.63×10^{-2}	9.72×10^{-2}

Table 4.3: Coefficient of variation (CoV) of Bayesian evidence methods

4.4.2 Comparison of Bayesian evidence methods

Using our established criteria for comparing the Bayesian evidence methods and the demonstrative example results, we now discuss each estimation approach, focusing on their strengths and weaknesses and applicability to Bayesian ARMA-GARCH model order choice.

1. Nested sampling

Metric	Conclusion
Accuracy	NS's effectiveness in handling complex, multimodal problems is due to its ability to focus sampling in areas where the likelihood is significant. This strength is evident in the demonstrative example results, where NS closely aligns with the baseline across all DoFs, suggesting high accuracy. This accuracy makes NS highly suitable for Bayesian evidence estimation in Bayesian ARMA-GARCH models, as it can reliably navigate the intricacies of financial time series data.
Scalability	While NS effectively identifies the bulk of the posterior mass, it can struggle in higher-dimensional spaces, a common scenario in complex financial models. The increased CoV in higher DoFs in the demonstrative example highlights this issue, suggesting reduced stability as dimensionality grows. Despite this, NS's relative stability in lower dimensions makes it a viable option for Bayesian evidence estimation in less complex Bayesian ARMA-GARCH models.
Efficiency	NS's iterative sampling from the prior distribution becomes increasingly challenging as the complexity and dimensionality of the parameter space grows. NS must maintain a representative set of samples (live points) that can capture the structure of the posterior distribution; so, the method inherently requires more evaluations in higher dimensions, where generating a new sample with a higher likelihood can be very difficult. The necessity for more samples in higher dimensions, as shown in the demonstrative example, indicates that while efficient for simpler models, NS might be too computationally demanding for Bayesian evidence estimation in highly complex Bayesian ARMA-GARCH models.
Complexity	The straightforward implementation of NS, requiring only sample generation from a prior distribution and likelihood evaluation, is advantageous. This simplicity makes NS an accessible method for Bayesian evidence estimation in Bayesian ARMA-GARCH models, especially if computational resources are limited.
Robustness.	The increasing variability with DoF suggests some sensitivity of NS to parameter changes in high-dimensional settings; however, the consistent performance across all DoFs in the demonstrative example indicates robustness. This robustness and consistent performance render NS a reliable method for Bayesian evidence estimation.

Table 4.4: Summary of nested sampling for Bayesian evidence estimation

Drawing on these evaluations, we now comment on whether nested sampling is suitable under the Bayesian ARMA-GARCH specific criteria.

Efficient sampling and handling high-dimensional spaces: NS effectively transforms complex multi-dimensional integrals into simpler one-dimensional integrals over 'prior mass', which is particularly beneficial for ARMA-GARCH models requiring navigating complex posterior distributions. NS exhibits increased variability and reduced stability in higher dimensions, as shown in the demonstrative example, presenting challenges in the high-dimensional spaces typical of ARMA-GARCH models. This implies that NS is capable but may require extra caution and additional computational resources for optimal performance, where the computational costs escalate with increased dimensionality.

Adaptability to tail events: NS's significant strength lies in its ability to sample in those areas of greatest importance, which aids in efficiency. However, it does make the method less likely to capture tail events, a crucial aspect of ARMA-GARCH models.

Nested sampling's direct estimation of marginal likelihood without relying on approximations and strength in exploring complex parameter spaces are useful for Bayesian evidence estimation of ARMA-GARCH models. However, it faces challenges in high-dimensional spaces, requiring careful calibration to maintain stability, high computational costs, and issues with tail risk modelling.

Metric	Conclusion
Accuracy	MMC is very effective at exploring the complete target distribution, guiding sampling towards underexplored areas—allowing for efficient sampling across the whole distribution (even multimodal). Despite slightly higher deviations in higher dimensions, the close alignment with the baseline across all degrees of freedom (DoF) in the demonstrative example highlights its accuracy. This high level of accuracy makes MMC particularly suitable for Bayesian evidence estimation in Bayesian ARMA-GARCH models, ensuring a comprehensive understanding of the data's distribution.
Scalability	MMC's approach to guiding sampling towards regions of interest and new intermediate distributions in each iteration contributes to its stability, a key factor in handling the variability of financial models. The low standard deviation across dimensions in the demonstrative example underscores this stability, unaffected by the increase in dimensionality. Such scalability makes MMC a robust choice for Bayesian evidence estimation for ARMA-GARCH models.
Efficiency	MMC's efficiency is contingent on the nature of the distribution; it excels in lower-dimensional spaces, where it can easily flatten the probability landscape but requires significantly more samples to explore high-dimensional distributions common in financial models. The diminished efficiency in higher dimensions, as evidenced in the demonstrative example, highlights MMC's challenges in more complex scenarios. While efficient for simpler Bayesian ARMA-GARCH models, MMC might be less practical for highly complex models due to increased computational demands.
Complexity	MMC can be challenging to implement, especially regarding generating samples from the target distribution. This complexity might pose challenges for practitioners using MMC in Bayesian evidence estimation for ARMA-GARCH models, especially those without extensive computational resources or specialised expertise.
Robustness	MMC's performance across different DoFs indicates reasonable robustness, although increased standard deviation in higher dimensions suggests some sensitivity to model complexity.

Table 4.5: Summary of multicanonical Monte Carlo for Bayesian evidence estimation

Drawing on these evaluations, we now comment on whether multicanonical Monte Carlo is suitable under the Bayesian ARMA-GARCH specific criteria.

Efficient sampling and handling high-dimensional spaces: MMC obtains accurate solutions with low MSE for high-dimensional problems, albeit with a large number of samples. Its flat-histogram sampling ensures proportional exploration of all areas of the parameter space, a highly beneficial feature for multi-parameter ARMA-GARCH models. However, the efficiency of MMC decreases in very high-dimensional scenarios, as evidenced in the demonstrative example, indicating a potential challenge for complex financial models.

Adaptability to tail events: MMC's comprehensive coverage of the parameter space, including its ability to effectively explore tail distributions, makes it particularly adept at capturing tail events. This adaptability is crucial for financial models like ARMA-GARCH, where extreme market events, though rare, can significantly impact model accuracy and risk assessment.

Multicanonical Monte Carlo, with its unique approach to Bayesian evidence estimation, exhibits promising capabilities in the context of ARMA-GARCH models. MMC excels in efficiently sampling the parameter space and capturing tail events, which is essential in financial time series analysis. However, its iterative nature can be computationally demanding, particularly for high-dimensional models. The critical aspect lies in carefully tuning weights and parameters, which requires expertise and computational resources. This aspect might pose a challenge in practical applications, particularly in scenarios with limited computational capabilities. We directly address this later in Chapter 8.

3. Bridge sampling

Metric	Conclusion
Accuracy	BS's accuracy hinges on the appropriate choice of the bridging function to connect the posterior distributions effectively. High accuracy demonstrated in the demonstrative example across all degrees of freedom (DoF) indicates BS's capability to provide precise estimations of Bayesian evidence. This level of accuracy makes BS particularly suitable for Bayesian evidence estimation in Bayesian ARMA-GARCH models, where accurate model selection is pivotal.
Scalability	The challenge in BS lies in maintaining stability as the complexity of the posterior distribution increases. Finding an appropriate bridging function becomes more challenging in high-dimensional spaces like those often encountered in financial models. In the demonstrative example, the increased CoV at higher DoFs highlight potential stability issues as dimensionality increases. Caution is advised in applying BS to highly complex Bayesian ARMA-GARCH models.
Efficiency	BS is efficient regarding the number of samples required, especially if the bridging function is well-optimised. This optimisation reduces the computational burden significantly, an essential consideration in time-sensitive financial analyses. The demonstrative example results show a significantly lower number of evaluations is needed compared to NS, especially at higher DoFs. BS's efficiency makes it an attractive method for Bayesian evidence estimation in ARMA-GARCH models, particularly where computational resources are limited or in less complex model structures.
Complexity	While conceptually straightforward, selecting a suitable bridging function can add a layer of complexity to BS. This complexity may pose challenges in implementing BS for Bayesian evidence estimation in ARMA-GARCH models.
Robustness	BS shows overall robustness across different DoFs, yet the high variability at the highest DoF suggests potential sensitivity in more complex parameter spaces. This variability is a concern for high-dimensional ARMA-GARCH models.

Table 4.6: Summary of bridge sampling for Bayesian evidence estimation

Drawing on these evaluations, we now comment on whether bridge sampling is suitable under the Bayesian ARMA-GARCH specific criteria.

Efficient sampling and handling high-dimensional spaces: BS's design focuses on efficient sampling from the posterior distribution, a critical requirement for capturing the full spectrum of parameter values in Bayesian ARMA-GARCH models. This efficiency is highlighted by its performance in the demonstrative example, requiring fewer evaluations than the vertical likelihood methods, particularly at higher DoFs. However, BS's effectiveness seems to wane in higher-dimensional settings, typical of more complex ARMA-GARCH models. The observed increase in variability at higher DoFs suggests a potential decrease in the method's efficiency and stability in these scenarios.

Adaptability to tail events: The ability of BS to effectively bridge the posterior distributions is advantageous in exploring the entire parameter space, including the tails. However, the increased variability noted at higher dimensions could impact BS's consistency in capturing these critical tail events.

Overall, BS emerges as a valuable method for Bayesian evidence estimation in financial time series models, especially those of moderate complexity and lower dimensionality. It demonstrates commendable accuracy and stability in such settings. However, for ARMA-GARCH models with high-dimensional parameter spaces, the increased variability and potential challenges in handling complex distributions warrant careful consideration. The choice of whether to use bridge sampling should be informed by the Bayesian model's specific requirements, balancing its strengths in efficient sampling and adaptability against the potential limitations in high-dimensional spaces and complex scenarios.

Metric	Conclusion
Accuracy	TI's high accuracy stems from its ability to systematically bridge between the prior and posterior distributions, capturing the nuanced variations in the parameter space. In our demonstrative example, the TI mean values are very close to the baseline, indicating high accuracy. The method aligns well with the true baseline solution, especially for low DoFs. This method is highly suitable for accurately estimating Bayesian evidence in Bayesian ARMA-GARCH models.
Scalability	TI demonstrates high stability due to its approach to transitioning between distributions. This characteristic is beneficial in managing the complexity of financial models, which often involve a wide range of parameters. The results display high stability with consistently low CoV across all degrees of freedom, suggesting little concern when dealing with high dimensional problems. Its consistent performance across different dimensions makes TI a reliable choice for Bayesian evidence estimation in simple and complex ARMA-GARCH models.
Efficiency	The efficiency of TI hinges on the well-set bridging distributions. While it requires a similar number of evaluations as other methods, its complex calculations can demand more computational time, a critical factor in time-sensitive financial modelling.
Complexity	The complexity of TI lies in establishing a series of distributions that effectively bridge the prior and posterior, requiring careful setup and understanding of the model's dynamics.
Robustness	TI uses a temperature schedule to bridge between the prior and posterior distributions to ensure that the most important parts of the parameter space contribute to the integral. The method is very sensitive to the choice of temperature and path. When the temperature and path do not fully capture the full complexity of likelihood distribution, the performance is poor, as seen in our demonstrative 'TI-Equal' example. This sensitivity necessitates careful calibration of the temperature schedule and path in TI, making it a method that requires meticulous setup. It may be less suitable when model parameters are frequently adjusted, like in highly dynamic financial environments.

Table 4.7: Summary of thermodynamic integration for Bayesian evidence estimation

Drawing on these evaluations, we now comment on whether thermodynamic integration is suitable under the Bayesian ARMA-GARCH specific criteria.

Efficient sampling and handling high-dimensional spaces: TI is capable of efficient sampling from the posterior distribution, a critical aspect for capturing the full range of parameter values in complex models like ARMA-GARCH. TI performs well even in high-dimensional settings, although this efficiency can be compromised by TI's sensitivity to the bridging strategy, necessitating careful tuning to ensure optimal performance.

Adaptability to tail events: TI's adaptive bridging process allows for a thorough exploration of the entire parameter space, including the tails. By effectively capturing these extreme but rare events, TI enhances the robustness of the model selection process.

In conclusion, TI exhibits commendable accuracy and potential for efficient sampling in complex financial models, especially with an adaptive bridging strategy. Its performance, however, tends to be less stable in higher-dimensional scenarios, underlining the importance of a meticulously designed bridging strategy. Although TI is complex and requires careful calibration, it remains a key method for Bayesian model selection in financial time series analysis. Its ability to handle complex likelihood surfaces and provide accurate marginal likelihood estimates is invaluable, particularly in scenarios where other methods might falter. The key to leveraging TI's full potential in ARMA-GARCH models lies in the optimal design and implementation of the bridging strategy.

Metric	Conclusion
Accuracy	SSS achieves high accuracy by employing a sequence of intermediate distributions that bridge the gap between the prior and posterior, allowing for a more thorough exploration of the parameter space; ensuring that the key features of the distribution are well-captured. This strength is underscored by its close alignment with the baseline in the demonstrative example, showcasing its ability to reflect underlying data dynamics accurately. The precision is highly beneficial for accurately estimating Bayesian evidence in ARMA-GARCH models.
Scalability	The stability comes from SSS's approach to transition between distributions, which minimises abrupt changes in the parameter space. This gradual transition ensures consistent performance, especially in lower-dimensional settings where the transition steps are more manageable. The low CoV in the demonstrative example demonstrates this stability. The stable performance of SSS is advantageous for Bayesian evidence estimation in ARMA-GARCH models, ensuring reliable results.
Efficiency	SSS's efficiency is attributed to its sequential approach, which does not necessarily require a drastic increase in computational effort as the dimensions increase. The efficiency stems from the ability to reuse calculations across steps, reducing the total computational load. The consistency in the number of evaluations required across different DoFs suggests an efficient use of computational resources. This efficiency is beneficial for Bayesian evidence estimation in ARMA-GARCH models, as it ensures thorough exploration without excessive computational demands.
Complexity	The complexity of SSS lies in defining a sequence of distributions between the prior and posterior, requiring careful setup and understanding of the model's dynamics. This process can be challenging—although methods exist for adaptive implementation.
Robustness	Issues arise in SSS's robustness if the power posteriors do not adequately cover the target distribution's full complexity due to the poor choice of intermediate distributions. The underestimation observed in the demonstrative example when these power posteriors are not comprehensive enough ('SSS-Equal') highlights this concern. While robust in general, careful construction of power posteriors is essential for SSS to be effective.

Table 4.8: Summary of steppingstone sampling for Bayesian evidence estimation

Drawing on these evaluations, we now comment on whether steppingstone sampling is suitable under the Bayesian ARMA-GARCH specific criteria.

Efficient sampling and handling high-dimensional spaces: SSS's approach of progressing through intermediate distributions enables an efficient exploration of the parameter space. This efficiency is particularly advantageous for complex models like ARMA-GARCH, where the need to navigate through a diverse range of parameter values is critical. The method ensures a thorough coverage of the parameter space without excessive computational demands, a significant factor in modelling time-sensitive financial data—even in high-dimensional spaces.

Adaptability to tail events: SSS's technique of bridging between the prior and posterior distributions allows it to explore the extremities, or tails, of the distribution. By capturing these events, SSS comprehensively estimates Bayesian evidence for ARMA-GARCH models.

Overall, SSS, particularly when employing an adaptive bridging strategy, presents a highly effective approach for Bayesian evidence estimation in financial time series models. Its capability to handle complex, high-dimensional spaces and adapt to tail events, combined with its efficiency and accuracy, makes it a strong candidate for ARMA-GARCH model analysis. However, the intricacies involved in its implementation and the need for careful design of the sequence of distributions can be challenging.

4.4.3 Conclusion

The comparison of various Bayesian evidence estimation methods, specifically contrasting vertical likelihood representations (such as nested sampling and multicanonical Monte Carlo) with non-parametric alternatives (including bridge sampling, thermodynamic integration, and steppingstone sampling), reveals insightful distinctions, particularly in the context of financial models like ARMA-GARCH.

Vertical likelihood methods: Nested sampling and multicanonical Monte Carlo stand out in their ability to navigate complex, high-dimensional parameter spaces. By fully exploring the posterior distribution, these methods can capture tail events. However, such approaches can be computationally intensive, especially in higher dimensions, which might limit their practicality in certain scenarios.

Non-parametric alternatives: Bridge sampling, thermodynamic integration, and steppingstone sampling offer more flexibility in handling the intricacies of financial models. Their approach, particularly in the case of steppingstone sampling, is characterised by a systematic progression through a sequence of intermediate distributions, allowing for a detailed exploration of the parameter space. While these methods are generally efficient and robust, they require careful calibration and are sensitive to the choice of bridging strategy, especially in high-dimensional spaces.

In conclusion, vertical likelihood and non-parametric methods have unique strengths and limitations in the context of Bayesian ARMA-GARCH model order selection. The choice between these methodologies should be guided by the model's specific requirements, considering factors like dimensionality, complexity, and the need for efficient sampling. With their flexibility and adaptive strategies, non-parametric methods might offer an edge in more complex, high-dimensional scenarios. In contrast, vertical likelihood methods could be more suitable for scenarios where computational efficiency is less important but capturing tail risk is.

4.5 Establishing the Bayesian model

ARMA-(T)GARCH models combine ARMA's proficiency in predicting the mean of a series with (T)GARCH's ability to model and forecast changing volatility, considering volatility clusters and asymmetry. A Bayesian approach allows for the incorporation of prior knowledge and direct consideration of uncertainty.

4.5.1 Key data analysed

For our core dataset, the significant ACF and PACF values, as presented in Table 2.6, serve as potential candidates for ARMA model orders. Using steppingstone sampling, we calculate the Bayesian evidence for each combination, including the additional values $\{1, 2, 3, 5, 8\}$. The complete Bayesian evidence results are provided in the table in Appendix B.1.

Additionally, we estimate the Bayesian evidence for each combination of potential TGARCH model orders—based on the ARMA residuals—using steppingstone sampling. These results are also detailed in Appendix B.1.

The optimal ARMA and TGARCH model orders, based on the Bayesian evidence, are presented in Table 4.9.

	ARMA orders		**TGARCH orders**		
Asset ticker	**p**	**q**	**m**	**t**	**s**
SP500	7	15	2	1	1
AAPL	9	0	3	1	9
ALK	0	17	3	2	10
FTSE100	9	0	1	1	1
SHEL	0	23	1	1	1
TW	5	0	2	3	3
SSE	8	15	1	3	2
PING	0	1	1	1	1
SHEN	3	12	6	3	4

Table 4.9: ARMA-TGARCH Model orders for key data

4.5.2 Achieving objective one

In conclusion, this chapter has been driven by our first thesis objective, to develop algorithms that transcend the limitations of traditional financial models, enabling a more accurate and robust prediction of asset returns and volatility considering the inherent uncertainty in financial markets. Bayesian methods offer a natural framework for quantifying uncertainty in ARMA and GARCH models.

By recognising that model order selection and parameter tuning are critical in pursuing more accurate forecasts, we explored various prior distributions for ARMA and GARCH parameters and approaches to fine-tune them. Finally, we compared Bayesian evidence estimation methods for choosing optimal model orders in ARMA and GARCH.

From the posterior distributions of the Bayesian ARMA-GARCH models, a probability distribution of future forecasted returns can be obtained. It remains to explore how to do so, in particular, to obtain expected return and tail risk measures. Focus now shifts to the second objective for 'enhanced risk management', which is to develop new methodologies that better predict the probability and impact of tail events.

Chapter 5

Financial Risk

Tail risk refers to financial risk arising from extreme market events that fall outside the typical range of expectations and historical norms. These events, often characterised as "Black Swan" events, can lead to significant deviations in asset prices and investment returns (Taleb, 2005). Quantifying tail risk is critical because these events, while low in probability, can have disproportionately large and often catastrophic impacts on investment portfolios. A prime example is the Global Financial Crisis (GFC) of 2007-08, which was triggered by the collapse of the subprime mortgage market in the USA and led to the subsequent failure of many major financial institutions. The GFC was largely unpredictable and far-reaching in its consequences: triggering a global economic downturn, significant losses in stock markets, widespread unemployment, and a decrease in consumer wealth (McKibbin and Stoeckel, 2010).

Historically, investor forecasting activity has focused on predicting frequent but relatively unimportant events, as shown by Bond and Dow (2021). Consequently, investors often only bet on or against frequent events, failing to account for tail risk. By quantifying and considering tail risk, investors can better prepare for unforeseen market disruptions and allocate their resources more effectively.

Harder than predicting the occurrence of a rare event is the complexity of predicting the wide-ranging, often inconsistent consequences. As an example, the Russia–Ukraine conflict that began in late February 2022 caused global equities to broadly experience a sharp decline from mid-February to mid-March 2022 (Federal Reserve Bank of St. Louis, 2022). However, some stocks experienced significant gains, like in the Oil and Gas sector, where Shell Plc's stock price rose 15% from March 2022 to March 2023 due to benchmark oil prices surging nearly 30% from before the conflict to a high on March 8, 2022. The war demonstrates how hard rare events are to assess probabilistically, further motivating the need to consider uncertainty in financial markets and mathematical models.

This chapter focuses on the quantitative measurement of market and credit risk[1], considering tail risk and uncertainty quantification—working towards the second thesis objective, to develop new methodologies which better predict the probability and impact of tail events.

5.1 Quantitative measurement of risk

Investments necessitate the acceptance of risk to surpass the risk-free rate of return[2] (RfR). Risky assets should be priced such that the risk premium aligns with the risk associated with the expected excess return (Bodie et al., 2002). The Sharpe ratio assesses the ratio of the excess expected return to the standard deviation of returns: Sharpe ratio $= \frac{\text{Expected Return - RfR}}{\text{Standard Deviation}}$; helping to compare different asset's performance and idiosyncratic risk. To mitigate idiosyncratic risk, investors can utilise

[1]The risk to an investment portfolio from movements in the market, such as equity & commodity prices, foreign exchange & interest rates, is called *Market risk*. The risk to debt securities that one counterparty does not fulfil their obligation, in part or full, on the agreed-upon date is called *Credit risk*, or *Default risk*.

[2]The risk-free rate of return is the theoretical rate of interest an investor would gain from a zero-risk investment over a specified period; typically represented by U.S. Treasury bill yields for USD investments and U.K. Gilt yields for GBP investments.

diversification to reduce the impact of any single asset's performance. However, diversification cannot eliminate systematic risk which affect the whole market.

From our core data, the Sharpe ratio and risk-return profiles of our individual assets can be compared to an equally-weighted portfolio[3]. The results show that the portfolio has a lower standard deviation for equivalent mean return than most individual assets, demonstrating the benefits of diversification.

Asset	**Daily net returns**	**Daily log returns**
Portfolio	0.06	0.05
SP500	0.04	0.04
AAPL	0.07	0.06
ALK	0.04	0.03
FTSE100	0.01	0.01
SHEL	0.01	0.01
TW	0.04	0.03
SSE	0.01	0.00
PING	0.03	0.02
SHEN	0.03	0.02

Table 5.1: Sharpe ratio of core data

[3]Using the historical data, the equally-weighted portfolio is constructed by allocating an equal amount of capital to each asset on January 1, 2010, and measuring the performance through time.

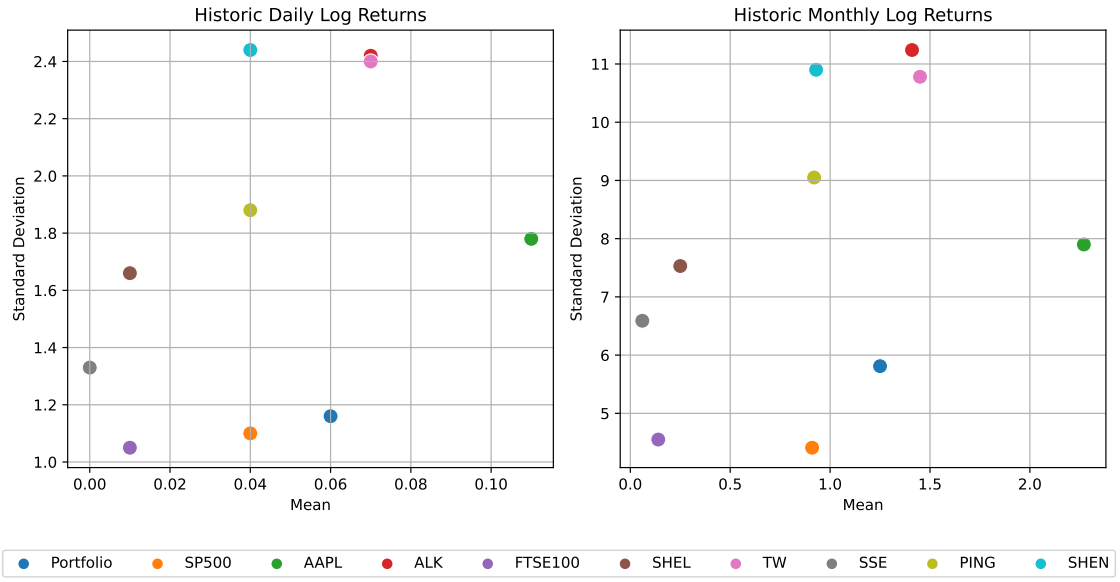


Figure 5.1: Risk return profile of core data

Whilst useful for its simplicity, standard deviation does not account for the return distribution's shape or the likelihood of extreme outcomes. The correlation between historical mean returns and standard deviations is weak—as seen in the plots—so such a metric does not suffice for comprehensive risk assessment. The Sharpe ratio falls short in accounting for the interdependencies between different assets. More advanced metrics are needed for tail risk management.

5.1.1 Return relationships

Covariance, the product of the deviations of two returns from their means, offers insights into the directional relationship between two assets. The sign indicates the nature of the relationship (positive for moving together, negative for moving inversely); its magnitude is less intuitive due to scale dependencies. Table 5.2 gives the covariances for our key data.

Correlation provides a normalised view, quantifying the strength of relationships between returns. A correlation near one suggests that the returns move in tandem,

indicating exposure to common risks. A correlation of zero implies there is no relationship. This understanding is pivotal, as assets with high positive correlations may not effectively diversify a portfolio's risk. Table 5.3 gives the correlations for our key data.

Table 5.2: Covariance of daily log returns

	SP500	AAPL	ALK	FTSE100	SHEL	TW	SSE	PING	SHEN
SP500	-	-	-	-	-	-	-	-	-
AAPL	1.30×10^{-4}	-	-	-	-	-	-	-	-
ALK	1.53×10^{-4}	1.41×10^{-4}	-	-	-	-	-	-	-
FTSE100	7.30×10^{-5}	6.40×10^{-5}	1.03×10^{-4}	-	-	-	-	-	-
SHEL	8.30×10^{-5}	6.40×10^{-5}	1.27×10^{-4}	1.32×10^{-4}	-	-	-	-	-
TW	9.60×10^{-5}	7.50×10^{-5}	1.56×10^{-4}	1.33×10^{-4}	1.17×10^{-4}	-	-	-	-
SSE	2.20×10^{-5}	3.00×10^{-5}	2.50×10^{-5}	2.90×10^{-5}	3.10×10^{-5}	4.10×10^{-5}	-	-	-
PING	3.70×10^{-5}	3.90×10^{-5}	4.80×10^{-5}	6.10×10^{-5}	6.20×10^{-5}	8.80×10^{-5}	1.32×10^{-4}	-	-
SHEN	2.10×10^{-5}	3.70×10^{-5}	3.30×10^{-5}	2.00×10^{-5}	1.70×10^{-5}	4.30×10^{-5}	1.64×10^{-4}	1.04×10^{-4}	-

Table 5.3: Correlation of daily log returns

	SP500	AAPL	ALK	FTSE100	SHEL	TW	SSE	PING	SHEN
SP500	-	-	-	-	-	-	-	-	-
AAPL	0.660	-	-	-	-	-	-	-	-
ALK	0.574	0.327	-	-	-	-	-	-	-
FTSE100	0.626	0.343	0.406	-	-	-	-	-	-
SHEL	0.453	0.215	0.317	0.755	-	-	-	-	-
TW	0.362	0.174	0.269	0.530	0.294	-	-	-	-
SSE	0.146	0.126	0.078	0.209	0.140	0.128	-	-	-
PING	0.177	0.117	0.106	0.307	0.198	0.195	0.527	-	-
SHEN	0.079	0.085	0.056	0.080	0.041	0.074	0.506	0.226	-

By examining the covariance and correlation matrices, one can identify how asset returns co-move, revealing both the magnitude and directionality of their relationships. In financial markets, investment returns have infinite tails, so correlation cannot be truly defined, as it relies on finite variance. Furthermore, correlation tells us very little about tail dependencies; therefore, a more sophisticated approach is warranted. Tail risk can be assessed by interpreting the distribution of forecasted returns using various risk measures—introduced next.

5.2 Equity risk

Equity risk encompasses the volatility and potential for loss due to fluctuations in stock prices driven by various market factors. This risk is particularly pronounced during extreme market events, which can cause significant deviations in stock performance, resulting in substantial financial impacts.

Key notation

For a portfolio, we define an N-dimensional vector $\mathbf{w} = (w_1, ..., w_N)$ to represent the capital allocation or *portfolio weights* to N assets. Each component w_i corresponds to the fraction of the total capital allocated to the i^{th} asset. The vector $\mathbf{w}$ is defined within the constraints of a feasible set $\mathbb{W} = \{\mathbf{w} \in \mathbb{R}^N | w_i \geqslant 0, \sum_{i=1}^{N} w_i \leqslant 1\}$, to ensure that the sum of all weights does not exceed the total available capital.

To represent the forecasted future returns for the portfolio, we define a random vector $\mathbf{Z}$. In a Bayesian ARMA-GARCH model, the posterior predictive distributions of returns form the random vector $\mathbf{Z} = (Z_1, ..., Z_N)$, where Z_i represents the random variable of forecasted returns for asset i. The return function $f(\mathbf{w}, \mathbf{z})$ represents the forecasted portfolio return for an allocation $\mathbf{w}$ and realisation $\mathbf{z}$ from $\mathbf{Z}$. As such, $f(\mathbf{w}, \mathbf{Z})$ is a random variable representing the forecasted return for a portfolio $\mathbf{w}$, obtained as the sum of the product of the portfolio weights and their

respective forecasted return distributions.

Through obtaining an accurate estimate of the distribution $y_f = f(\mathbf{w}, \mathbf{Z})$, one can obtain the future expected return and measure of tail risk for a specific allocation $\mathbf{w}$; under uncertainty. The subsequent chapters will explore how this distribution can be obtained. First, the key risk measures using y_f are introduced.

For clarity: $f(\mathbf{0}, \mathbf{Z}) = 0$, as no capital invested means no returns; $f(\mathbf{w}, \mathbf{z}) < 0$ if the portfolio is forecasted to lose money and $f(\mathbf{w}, \mathbf{z}) > 0$ if the portfolio is forecasted to gain money. As an example, $f(\mathbf{w}, \mathbf{z}) = 0.1$ means a forecasted gain of 10%; $f(\mathbf{w}, \mathbf{z}) = -0.2$ means a forecasted loss of 20%.

5.2.1 Value-at-risk

It is possible to calculate a risk measure every day for each market variable; however, this does not provide a measure of the total risk across a portfolio. Value-at-risk (VaR) and conditional VaR (CVaR) exist to provide a single number that summarizes the total risk in a portfolio.

Till Guldimann developed VaR at J.P. Morgan in the late 1980s as a market risk management tool (Holton, 2016). VaR measures the worst loss over a given horizon under normal market conditions at a given confidence level (Jorion, 2000).

Definition 5.2.1 *The VaR of $f(\mathbf{w}, \mathbf{Z})$ at a specified risk level $\alpha \in (0, 1)$, denoted as $v_f(\mathbf{w}; \alpha)$, is defined as the threshold value ω such that the probability of the loss exceeding ω is at most $(1 - \alpha)$. Formally, VaR is defined as:*

$$v_f(\mathbf{w}; \alpha) = \inf\{\omega : \mathbb{P}(f(\mathbf{w}, \mathbf{Z}) \leqslant -\omega) \leqslant 1 - \alpha\} \tag{5.2.2}$$

The VaR confidence level α is typically set to a high value (e.g., 0.95 or 0.99) to assess the risk of extreme losses.

As an example, if a portfolio's 10-day VaR for a 99% confidence level ($\alpha = 0.99$)

is 0.2, there is a 99% probability that the portfolio will **not** lose more than 20% of its value in the next 10 days. The remaining 1% represents the uncertainty, where losses could exceed the VaR threshold.

VaR does not account for the size of potential losses beyond the VaR threshold in the tail of the distribution, as such it is an incomplete measure of risk. While VaR answers the question, "How bad can things get?", it does not provide insights into "What happens beyond that point?" (Bodie et al., 2002).

5.2.2 Conditional value-at-risk

To address this limitation, conditional value-at-risk (CVaR) measures the expected loss during an N-day period conditional on the losses being worse than the VaR value. It is a measure for the worst-case scenario, asking how much value an investment portfolio is expected to lose, given the VaR threshold is met.

Definition 5.2.3 *The CVaR of $f(\mathbf{w}, \mathbf{Z})$ at a specified risk level $\alpha \in (0, 1)$ is defined as the expected loss, assuming that the loss is worse than the VaR threshold. It represents the average of the worst-case losses. Formally, CVaR is defined as:*

$$CVaR_\alpha[f(\mathbf{w}, \mathbf{Z})] = -\mathbb{E}[f(\mathbf{w}, \mathbf{Z}) | f(\mathbf{w}, \mathbf{Z}) \leqslant -v_f(\mathbf{w}; \alpha)] \tag{5.2.4}$$

where $v_f(\mathbf{w}; \alpha)$ is the VaR, as defined previously.

For example, if the CVaR at the 99% confidence level ($\alpha = 0.99$) is 0.35, then the average losses under the worst 1% of scenarios are expected to be 35%. This additional insight better considers the tail end of the loss distribution.

Considering two portfolios, with return distributions as shown in Figure 5.2, the difference between VaR and CVaR can be visualised. Both portfolios would have the same VaR but different CVaR—demonstrating the benefits of CVaR in tail risk

measurement.

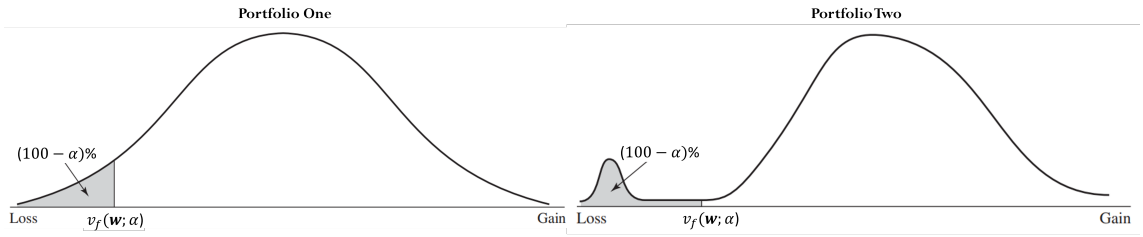


Figure 5.2: VaR and CVaR for two portfolios (adapted from Hull (2012))

5.2.3 Alternative risk measures

The widely cited paper by Artzner et al. (1999) establishes key desirable properties for risk measures. Of those, CVaR meets many, including subadditivity, translation invariance, positive homogeneity, and monotonicity. In contrast, VaR lacks sub-additivity, convexity, and differentiability. Beyond VaR and CVaR, various alternative risk measures have been developed to address different aspects of risk management—as compared in the work of Fischer et al. (2018). This work is summarised below, considering each metric and its strengths and weaknesses.

An additional measure, not in the Fischer analysis, is Liquidity-adjusted VaR (LVaR). The main objective of LVaR is to account for liquidity issues during market downturns. These issues make selling assets harder, exposing the investor to greater risks. The first version of LVaR was proposed by Angelidis and Benos (2006) in 2006. The method uses a bid-ask spread to adjust the estimated sale price of an asset and, consequently, the VaR figure. Another approach is suggested by Al Janabi et al. (2019), part of a portfolio optimisation framework that uses historical close-out periods for different asset classes to modify the VaR calculation.

Risk measure	Description	Advantages	Disadvantages	Coherence	Convex	Comonotonic add.	Law invariant	Elicitability	Robustness
Value-at-risk (VaR)	VaR is a statistical measure that quantifies the level of financial risk over a specific time frame. VaR represents the maximum expected loss over a given time period for a specified confidence level.	Easy to calculate and understand.	Not coherent as it fails to account for the magnitude of losses beyond the VaR threshold; not subadditive, which can discourage diversification.	×	×	✓	✓	✓	Weak Topology
Conditional value-at-risk (CVaR)	CVaR is the expected loss beyond the VaR level.	Coherent risk measure that accounts for tail losses; encourages diversification; and gives information about the size of extreme losses.	More complex to calculate than VaR; can be sensitive to assumptions about tail behaviour of the loss distribution.	✓	✓	✓	✓	×	Wasserstein

Median Shortfall (MS)	Variation on CVaR, using the median of the α-tail distribution, so both frequency and severity of extreme losses beyond VaR are considered.	Elicitable, robust with respect to the weak topology, and easy to implement.	Shares the same drawbacks as VaR since $MS_\alpha(L) = VaR_{\frac{1+\alpha}{2}}(L)$.	×	×	✓	✓	✓	Weak Topology
Expectile VaR (ExVaR)	Uses an asymmetric quadratic optimisation problem and least squares estimation to consider the entire distribution (unlike quantiles). Provides a better estimate around the α-threshold.	Only known coherent and elicitable alternative. More conservative estimate for heavy-tailed distributions, so better captures large losses.	Depends on the whole distribution, so the complete loss function must be reconstructed. Not comonotonic additive, which can lead to misleading diversification incentives.	✓	✓	×	✓	✓	Wasserstein
Lambda VaR (LVaR)	Works by narrowing the loss distribution range from which VaR is calculated, controlling the relationship between the probability and severity of losses through a probability function.	Monotone and law invariant, with properties of elicitability and robustness.	LVaR is not translation invariant and, therefore, cannot be a monetary risk measure nor optimisation purposes.	×	×	×	✓	✓	C-robust

VaR on benchmark loss distributions (BLDs)	VaR based on BLDs which take account of both the frequency and severity of losses, aiming to standardise risk measurement across different portfolios or firms by using a common set of loss scenarios.	Facilitates comparison across different portfolios or firms by using a standard set of loss scenarios; can provide a more uniform approach to risk assessment in diverse market conditions.	Lacks flexibility as it is tied to specific benchmark distributions. Like VaR, it does not account for the magnitude of losses beyond the VaR threshold and is not a coherent risk measure. Not: convex, subadditive, positive homogeneous, or elicitable.	×	×	×	✓	×	C-robust
Range VaR (RVaR)	A robust alternative to CVaR in cases where the range of loss probabilities vanishes around 0 and 1. RVaR calculates the average of VaR levels across a predefined range.	Robust in cases where the range of loss probabilities vanishes around 0 and 1, whereas CVaR would not be.	Not a coherent risk measure in general.	×	×	✓	✓	×	C-robust

Wang	Transformation-based risk measure that adjusts the probability distribution of losses using a distortion function, which skews the distribution to give more weight to tail events, emphasising extreme losses.	Unlike VaR and CVaR, Wang takes into consideration all available information from the loss distribution, providing a more comprehensive view of risk by focusing on extreme loss scenarios.	The choice of distortion function can significantly impact the measure, introducing subjectivity; more complex to implement and understand compared to simpler measures; may not always provide intuitive results.	✓	✓	✓	✓	×	-
Glue VaR	Four-parameter family of risk measures which allows for different measures to be calculated from one equation (including (C)VaR). Can be calibrated to be less or more conservative.	Can alter parameters to see how the risk profile changes based on how conservative an estimate you want.	Complex to implement compared to VaR and CVaR.	×	×	✓	✓	×	Wasserstein
Entropy Risk Measure (ERM)	Tightest upper bound of (C)VaR using Chernoff's inequality. Considers the set of payoffs with positive expected utility.	More efficient computation than CVaR, due to ERM being strongly and strictly montone. Also, coherent, convex, comonotonic additive, and law invariant.	Computational complexity in scenarios involving non-linear or complex loss distributions; challenging to implement and interpret in practical risk contexts.	✓	✓	✓	✓	×	-

5.2.4 Essential properties of CVaR

VaR often exhibits multiple local extrema and unpredictable behaviour as a function of portfolio positions, limiting its usefulness in portfolio optimisation problems, limitations which do not apply to CVaR (Mausser and Rosen, 1999; McKay and Keefer, 1996). Additionally, to achieve this thesis's central objective of developing a new investment framework that can handle and account for uncertainty, a key aspect is ensuring the robustness of the chosen risk measure to uncertainty. Embrechts et al. (2022) published a framework for considering the effect of uncertainty around an α-quantile level, concluding that, unlike VaR, CVaR remains stable and robust in simulation-based optimisation methods with uncertainty.

The described features and resilience to uncertainty position CVaR as the risk measure we choose to base our investment framework and later portfolio optimisation algorithm on[4]. As CVaR is the central risk measure used in the subsequent research & chapters, it is imperative to understand the properties beyond those explained so far.

Stochastic dominance of CVaR

Stochastic dominance is used to compare two probability distributions: if one distribution can be considered more favourable than another under certain conditions, it is said to *stochastically dominate* the other. CVaR is monotonic with respect to stochastic dominance of order 1 and order 2 (Pflug, 2000). In financial risk management, this property is essential. Let us formalise the first case:

[4]The significance of CVaR extends to financial regulations too. The transition from Basel III to Basel IV Banking regulations requires a shift from VaR to CVaR in Bank's Economic Capital (ECAP) calculations. ECAP is based on a bank's risk profile, aimed at preventing liquidity or insolvency crises (Saissi Hassani and Dionne, 2021).

Claim 5.2.5 *(Pflug, 2000) For stochastic dominance of order 1, if a random variable X_1 stochastically dominates X_2, then:*

$$CVaR(X_1) \leqslant CVaR(X_2). \tag{5.2.6}$$

Proof. For stochastic dominance of order 1, X_1 stochastically dominates X_2 if for every x, the cumulative distribution function $F_{X_1}(x) \leqslant F_{X_2}(x)$. Therefore, X_1 has a distribution skewed towards higher values compared to X_2. For a given level of α, the VaR and consequently the CVaR for X_1 will be lower than or equal to that of X_2. □

This property implies that when comparing two investment options, if one demonstrates lower risk (as indicated by CVaR) while providing equal or higher expected returns, it is universally more favourable regarding the risk-return balance—as expected in the practical context. This enables us to confidently use CVaR as a reliable and robust risk measure within portfolio optimisation.

CVaR-return tradeoff

As discussed in Chapter 1, the underlying principle in investing is that a riskier asset should generate higher returns to compensate investors for taking on greater risk. A robust risk measure should reflect this feature. Research by Bali and Cakici (2004) and Iqbal et al. (2010) proved a positive relationship between CVaR and expected returns. To visualise this relationship, the historical daily returns (from our core data) are plotted against the VaR and CVaR in Figure 5.3.

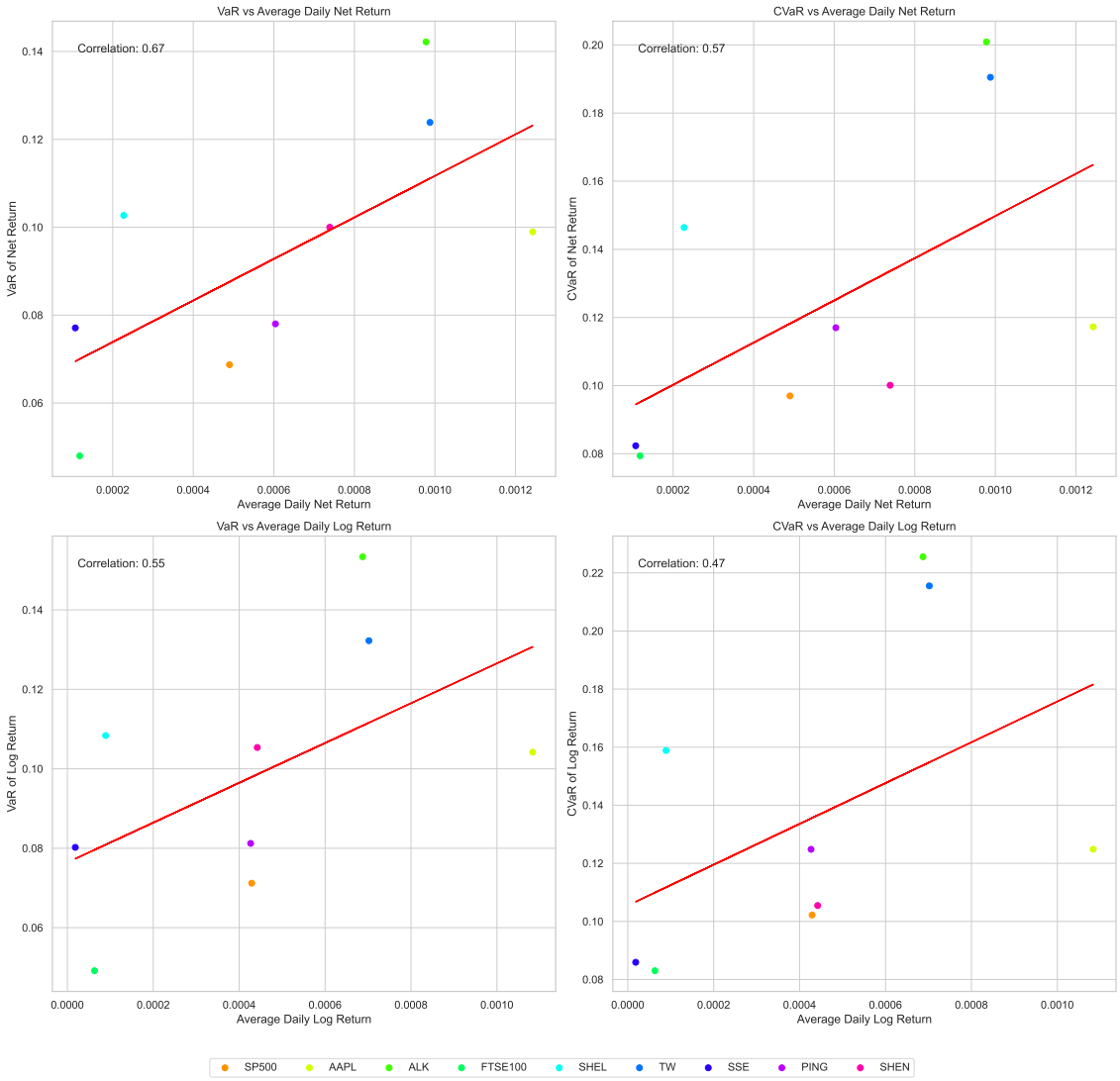


Figure 5.3: VaR and CVaR against average returns from historic data

A higher expected return can only be obtained by increasing risk exposure; conversely, if an investor wishes to reduce the CVaR of their portfolio, the expected return will reduce too.

Stochastic Dominance means that for a chosen expected return, an investor can identify a portfolio that meets or exceeds this return with the lowest possible risk, as measured by CVaR. When optimising a portfolio, these concepts imply that to achieve a certain level of expected return, one must consider the associated CVaR.

The optimal portfolio, in this context, would be the one that offers the desired expected return with the lowest possible CVaR. If the goal is to reduce the portfolio's CVaR, the expected return will decrease too. This relationship lays the groundwork for the portfolio optimisation algorithm developed in Chapter 10.

CVaR calculation

Three primary methods are used to calculate VaR and CVaR: the historical method, the variance-covariance method, and Monte Carlo simulations. Nadarajah et al. (2014) provides an extensive comparison of 45 different methods for estimating CVaR and concludes that Monte Carlo simulations offer the greatest accuracy and consistency, particularly during economic downturns. For our Bayesian ARMA-GARCH models, the ability of VaR and CVaR to capture tail risk is dependent on the ability to reconstruct the distribution $f(\mathbf{w}, \mathbf{Z})$. In the following chapters, we detail such simulation methodologies.

This concludes our introduction to market risk. Whilst the core data used throughout this thesis is based on equities, our research would be incomplete without consideration of credit risk.

5.3 Credit risk

In fixed-income debt securities, *coupon payments* are the periodic interest payments made to bondholders. The *maturity date* marks the end of the bond term when the *principal amount* is due to be repaid. A *default* refers to the failure of the borrower to meet these obligations; the associated risk is encapsulated in credit ratings. These ratings, provided by agencies like S&P and Moody's, serve as a barometer for default risk and significantly influence the market value of bonds.

Credit risk, or default risk, is a pivotal component of financial risk management. The essence of credit risk management lies in assessing the likelihood of default,

which can manifest as missed coupon payments or the inability to return the principal at maturity. Credit risk evaluation is influenced by numerous factors ranging from individual borrower characteristics to macroeconomic conditions.

In a portfolio of fixed-income assets, one could consider each asset's default risk separately, which is relatively straightforward. However, such an approach fails to account for the interdependencies between different securities and issuers, which manifests itself through the clustering of defaults often seen during periods of financial turmoil.

Default correlation describes the propensity for two or more companies to default at about the same time. Default correlation is caused by macroeconomic factors (e.g., recessions and regulatory changes) and credit contagion, where one company's financial difficulties directly affect other companies they are closely associated with. The interconnected nature of financial markets means that stresses can propagate quickly, with defaults in one area leading to a chain reaction of credit events and a clustering of external risk factors during market downturns—a phenomenon termed *extremal dependence*.

Copula models were developed to estimate this tail risk more accurately by allowing for the separation of a portfolio's dependence structure from each asset's marginal densities—representing the individual risks of each obligor—which can have different probability distributions. As such, copula models join or 'couple' the individual default risks of various fixed-income assets to consider the portfolio as a whole. By doing so, copula models capture the likelihood of simultaneous defaults, thus offering a more comprehensive measure of the potential for extreme losses in a portfolio.

5.3.1 Normal copula model

Before the Global Financial Crisis, the most widely used copula model was the 'Normal copula model' proposed by David Li at J.P. Morgan in 1997 (Li, 2000; Morgan et al., 1997).

The model

Consider a portfolio of loans consisting of N obligors. Suppose the probability of default for the ith obligor over the considered time horizon is $p_i \in (0,1)$, for $i = 1, ..., N$, and that if the ith obligor defaults, a fixed and given loss of c_i monetary units occurs.

A copula model introduces a vector of underlying latent variables $\mathbf{X} = (X_1, ..., X_N)$ such that the i^{th} obligor defaults if X_i exceeds a given threshold level x_i. This threshold x_i is set according to the marginal default probability of the ith asset so that $\mathbb{P}(X_i > x_i) = p_i$. The portfolio loss from defaults is given by

$$L(\mathbf{X}) = c_1 I_{\{X_1 > x_1\}} + ... + c_n I_{\{X_N > x_N\}}, \tag{5.3.1}$$

where $I_{\{X_i > x_i\}}$ denotes the indicator function, equal to 1 if $X_i > x_i$ and 0 otherwise. As per Glasserman and Li (2005) and Li (2000), in the Normal copula model the joint distribution of the latent variables $\mathbf{X} = (X_1, ..., X_N)$ follows a multivariate Gaussian distribution given by:

$$X_i = w_{i1} Z_1 + ... + w_{id} Z_d \ + \ w_i \eta_i, \tag{5.3.2}$$

where $w_{i1}^2 + ... + w_{id}^2 + w_i^2 = 1$; $(Z_1, ..., Z_d)$ are i.i.d. standard Normal random variables known as *common risk factors* that capture the systemic risks common to all the obligors; and η_i is a Normal random variable—independent of the common

risk factors—that captures the *idiosyncratic risk* specific to the ith obligor.

In summary, in the Normal copula model, each obligor's default risk is modelled by a latent variable X_i, which comprises both systematic risk factors $Z_1, \ldots, Z_d$ and an idiosyncratic risk factor η_i. The weights $w_{i1}, \ldots, w_{id}$ quantify the influence of each systematic risk factor on the ith obligor's default risk, while w_i represents the idiosyncratic risk's contribution. This model adeptly captures the shared risks that affect all obligors and the unique risks to individual obligors, thus evaluating the potential for concurrent defaults due to systemic events and singular defaults arising from idiosyncratic events.

Limitations

The Normal copula model—used by nearly every bank globally prior to the Global Financial Crisis (GFC) for risk management and asset valuation purposes, is now referred to as *The Formula That Killed Wall Street* due to its role in exacerbating the GFC (Bass, 2009). This model and its impact on the GFC is a powerful reminder of how the failure of institutions to build appropriate mathematical models and simulations can have deep consequences.

As Frey and McNeil warned in their 2001 paper (Frey and McNeil, 2001), as a result of the risk factors in the Normal copula model being modelled as multivariate-normally distributed random variables, which are asymptotically independent, the occurrence of many joint large movements of the risk factors is very rare. This contradicts reality, as large movements of many risk factors occur with non-negligible frequency. Frey and McNeil add that "this [key assumption] casts some doubt on whether latent variable models based on a Gaussian copula are necessarily the best choice for modelling dependent defaults".

The paper by Zeevi and Mashal (2002) added to this criticism, where following analysis of Gaussian dependence structures, compared to t-dependence structures,

the paper concluded that "the Gaussian dependence hypothesis that underlies most modern financial applications should be deemed inadequate", adding "[we] believe that the t-dependence structure is an important step towards a more realistic model of dependencies between underlying assets"—a model we explore next.

Before the GFC, most banks used the Normal copula model in risk management. As a result, they seriously underestimated tail risk & the likelihood of a large number of simultaneous asset defaults. This resulted in banks having insufficient capital and inappropriate risk management, exacerbating the crisis; hence, the Normal copula model gained its reputation as *The Formula That Killed Wall Street*. Much research has gone into developing better copula models, including the aforementioned Student's t Copula to better account for extremal dependence.

5.3.2 Student's t copula model

To address the weaknesses in the Gaussian copula model, Bassamboo et al. (2008) proposed an alternative model building on the work of Frey and McNeil (2001), replacing the Gaussian copula with a multivariate Student's t copula model. The Student's t distribution is similar to the Normal distribution but features heavier tails, which increase the likelihood of extreme values. Furthermore, the Student's t copula introduces a common random variable that amplifies the joint likelihood of extreme values occurring simultaneously across multiple variables, effectively capturing the dependencies between them.

The performance of the Student's t copula model is demonstrated in a review paper by Patton (2012), which assessed the performance of the Normal, Clayton, Rotation Gumbel and Student's t copula models using various tests. The Student's t model outperformed all other models when applied to four different goodness-of-fit tests.

The model

The loss function and set-up of the copula model, except for the Latent Variable equations, are the same as for the Normal copula model, presented in Section 5.3.1.

As per Bassamboo et al. (2008), the Student's t copula model specifies the joint distribution of the latent variables $\mathbf{X} = (X_1, ..., X_N)$, by modelling the dependence among obligors with latent variable following a multivariate Student's t distribution given by:

$$X_i = \frac{w_{i1}Z_1 + ... + w_{id}Z_d + w_i\eta_i}{T}, \tag{5.3.3}$$

where $w_{i1}^2 + ... + w_{id}^2 + w_i^2 = 1$; $(Z_1, ..., Z_d)$ are i.i.d. standard Normal random variables known as *common risk factors* and η_i is a Normal random variable that captures the *idiosyncratic risk* specific to the ith obligor. T is a random variable which takes value according to

$$T = \sqrt{k^{-1}\ \text{Gamma}(1/2, k/2)}\ , \tag{5.3.4}$$

an adapted Gamma distribution with shape $k/2$ and scale $1/2$. As such, the distribution given in Eq. 5.3.4 results in latent variables $(X_i : 1 \leqslant i \leqslant N)$ following a multivariate Student's t distribution, whose dependence structure is given by a t-copula with k degrees of freedom.

The random variable T introduces an additional layer of dependency among obligor defaults by representing common shocks. When T takes values close to zero, all the latent variables X_i are likely to exhibit large values simultaneously, leading to many simultaneous defaults. This mechanism allows the t-copula model to better capture tail risk and the clustering of extreme events.

Beyond merely modeling extreme events, the Student's t copula captures dependencies between variables through the concept of tail dependence—that is, the increased likelihood that extreme values in one variable will coincide with extreme

values in another. In the Student's t copula model, the latent variables $\mathbf{X} = (X_1, \ldots, X_N)$ are influenced by both common risk factors $(Z_1, \ldots, Z_d)$ and idiosyncratic risk factors η_i. The common risk factors create a correlation between variables, while the random variable T scales these correlations, particularly under extreme conditions. As T varies, it induces a dependency structure that allows for both linear and non-linear relationships between variables, capturing the complex nature of financial markets.

Theoretical analysis by Embrechts et al. (2001) demonstrates the upper and lower tail dependence properties of the t-copula model, confirming its ability to account for the co-movement of variables during extreme market conditions. This enhanced dependency modeling is why the Student's t copula is considered more effective in financial risk management compared to the Gaussian copula.

Further adaptations

Tang et al. (2019) present an adaptation to the copula model, which introduces a single common shock variable. This shock variable represents a significant market or economic event to see how a portfolio would perform. They conclude (somewhat intuitively) that large portfolio losses can be attributed to either the common shock variable or the systematic risk factor, whichever has a heavier tail.

Bassamboo et al. (2008) propose replacing the default thresholds x_i in the copula model with a function $f_i(n)$—for portfolio size n. Whereby $f_i(n)$ increases at a subexponential rate, as n increases, to account for portfolio diversification. So, higher diversification reduces the likelihood of large losses.

5.3.3 Copula model summary

The Student's t copula model marks a significant advancement in the ability to capture dependencies and tail risk within a credit portfolio. The Student's t copula

model relies on the ability to explore the parameter space in order to reconstruct the loss function—in subsequent chapters, we investigate how to do this.

5.4 Failure probability

Failure probability estimation is widely researched in engineering and other scientific disciplines to determine the likelihood and impact of undesirable outcomes or 'failures' in complex systems. By considering market risk measures (like VaR and CVaR) and large simultaneous defaults from credit risk models as *failures*, we can develop a unified framework for quantifying the likelihood of undesirable financial outcomes. By defining 'failure' as the event where financial outcomes, such as losses, exceed a predefined critical threshold, we can establish a common framework for developing and evaluating risk assessment methodologies.

5.4.1 The framework

To ensure clarity in the remaining presentations, we separate the notation used here for generic applications—to explain key methods, algorithms, and approaches—from the notation used previously that relates specifically to portfolio optimisation, forecasting asset returns and financial risk management.

Let $\mathbf{x}$ be a d-dimensional random vector following distribution $p(\mathbf{x})$; for clarity, $\mathbf{x} = (x_1, ..., x_d)$, where the i^{th} component of this vector is denoted by x_i. Let y_g be a scalar variable characterised by a function $y_g = g(\mathbf{x})$, called the *performance variable*, with respective probability density function $\pi(y_g)$. $\mathbf{x}$ and y_g are assumed to be continuous random variables.

The failure probability can be estimated by determining the performance variable's distribution $\pi(y_g)$, where a system failure event occurs when the performance

variable y_g exceeds a threshold value y_g^*—formalised as a set F:

$$F = \{\mathbf{x} \in \mathbb{X} |\ y_g = g(\mathbf{x}) > y_g^*\}. \tag{5.4.1}$$

The failure probability is obtained using the integral:

$$P_F = \mathbb{P}(F) = \int\limits_{\{\mathbf{x}\in\mathbb{X}|g(\mathbf{x})>y_g^*\}} p(\mathbf{x})d\mathbf{x}. \tag{5.4.2}$$

This can be more easily obtained using:

$$P_F = \mathbb{P}(F) = \int\limits_{\mathbf{x}\in\mathbb{X}} I_F(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \tag{5.4.3}$$

where $I_F(\mathbf{x})$ is defined as an indicator function of the set F:

$$I_F(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in F; \\ 0, & \text{if } \mathbf{x} \notin F. \end{cases} \tag{5.4.4}$$

To accurately estimate the failure probability, the distribution $\pi(y_g)$ must be obtained.

5.5 Distribution reconstruction

To achieve our second objective of developing new methodologies which better predict the probability and impact of tail events, it is essential to utilise sophisticated risk measures.

The assessment of equity risk requires obtaining the entire distribution of forecasted asset returns to calculate CVaR and quantify tail risk. In credit risk, the ability to accurately model and reconstruct the entire joint default distributions in

copula models is essential for evaluating the tail risk in a portfolio, as it determines the likelihood of extreme loss scenarios that can arise from clustered defaults.

The unified failure probability framework directly relies on the ability to reconstruct probability distributions of performance variables. In this context, a failure event is defined by the performance variable exceeding a certain threshold. While point estimates can provide a snapshot of the likelihood of failure, the full probability distribution of the performance variable is necessary to understand the range and severity of potential failures. This distributional perspective is crucial for comprehensive risk assessment, encompassing the frequency and severity of failure events.

In summary, all three areas – equity risk, credit risk, and failure probability estimation – necessitate not only the computation of point estimates but also the detailed reconstruction of probability distributions. This is essential for capturing tail risks and the inherent uncertainties in financial markets.

In the next two chapters, we explore the application of Monte Carlo methods and more sophisticated simulation techniques for failure probability estimation and, as such, CVaR estimation for equity market risk and the Student's t copula model for credit risk. This motivates the development of our own algorithm for probability distribution reconstruction.

Chapter 6

Monte Carlo Methods for Risk Analysis

Working towards our second thesis objective for enhanced risk management to develop new methodologies which better predict the probability and impact of tail events, we seek to determine the distribution of the performance variable $y_g = g(\mathbf{x})$—denoted by $\pi(y_g)$—from which failure probabilities and other risk measures can be estimated.

The most widely used rare event simulation[1] is the Monte Carlo (MC) method proposed by Stanislaw Ulam as part of his work on the Manhattan project, developing nuclear weapons in the late 1940s. The main strength of MC is its robustness; that is, the accuracy of the MC approximation does not depend on the dimensionality of the input space. So, unlike other methods (e.g. numerical integration), it does not suffer from the "curse of dimensionality". Whilst MC is robust, it is highly inefficient in estimating small probability due to the substantial number of samples required.

As a result, MC adaptations have been developed, the most popular being im-

[1]The field of *rare event simulation* is concerned with developing models to simulate events with extremely small probabilities, generally of the order 10^{-6} or less.

portance sampling (IS), which aims to increase the accuracy and efficiency of MC by using alternative biasing distributions that drive sampling towards the region of interest. In this chapter, we present the Monte Carlo method and importance sampling—which lays the groundwork for the more advanced methods which follow.

All rare event simulation methods rely on the generation of representative samples from target probability distributions; the most widely used methods are the Metropolis-Hastings algorithm (Hastings, 1970; Metropolis et al., 1953), which utilises Markov chains to generate samples following a target distribution $\pi(\cdot)$, the sequential Monte Carlo sampler (Del Moral et al., 2006), which generates samples from a sequence of distributions $\{\pi_t(\cdot)\}_{t=1}^{T}$, and population Monte Carlo. The chapter concludes with a comparison of these methods.

6.1 The Monte Carlo method

6.1.1 General framework

Suppose we have a target density $p(\mathbf{x})$ defined on a high-dimensional space $\mathbb{X}$, where $\mathbf{x}$ is a random vector defined on $\mathbb{R}^d$ and $g(\cdot)$ is a real-valued function defined on $\mathbb{R}^d$. A Monte Carlo simulation can approximate the expectation of $y = g(\mathbf{x})$ given by:

$$\overline{y} = \mathbb{E}[y] = \int_{\mathbf{x}\in\mathbb{X}} g(\mathbf{x})p(\mathbf{x})d\mathbf{x}. \tag{6.1.1}$$

The Monte Carlo simulation works by drawing M independent identically distributed (i.i.d.) samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ from the distribution $p(\mathbf{x})$, computing $g(\mathbf{x}_j)$ for each $\mathbf{x}_j$ and then—by utilising the strong law of large numbers—estimating $\overline{y}$ using:

$$\overline{y} \approx \widehat{y} = \frac{1}{M}\sum_{j=1}^{M} g(\mathbf{x}_j). \tag{6.1.2}$$

6.1.2 Monte Carlo for failure probability estimation

As defined in Eq. 5.4.1, the probability of a failure event F is defined by:

$$P_F = \mathbb{P}(F) = \int\limits_{\{\mathbf{x} \in X | g(\mathbf{x}) > y^*\}} p(\mathbf{x}) d\mathbf{x} = \int\limits_{\mathbf{x} \in \mathbb{X}} I_F(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \tag{6.1.3}$$

where $I_F(\mathbf{x})$ is an indicator function of the set F, defined as

$$I_F(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in F; \\ 0, & \text{if } \mathbf{x} \notin F. \end{cases} \tag{6.1.4}$$

For simple low-dimensional problems, P_F can be estimated via standard Monte Carlo, drawing M samples $\mathbf{x}_1, ..., \mathbf{x}_M$ from the distribution $p(\mathbf{x})$ and estimating P_F with the following formula:

$$P_F \approx \frac{1}{M} \sum_{j=1}^{M} I_F(\mathbf{x}_j). \tag{6.1.5}$$

6.1.3 Monte Carlo for probability distribution construction

One approach to using the Monte Carlo method to construct a probability distribution $\pi(y)$, is to "bucket" the MC samples. For convenience, assume that $\pi(y)$ has bounded support[2] $R_y = [a, b]$, then decompose R_y into K bins of equal width Δ centred at the discrete values $\{b_1, ..., b_K\}$, where the k^{th} bin is defined as the interval $B_k = [b_k - \Delta/2, b_k + \Delta/2]$. This binning implicitly defines a partition of the input

[2]If the support is not bounded, choose an interval $[a, b]$ that is sufficiently large so that $\mathbb{P}(y \in [a, b]) \approx 1$.

space into K domains $\{D_k\}_{k=1}^K$, where

$$D_k = \{\mathbf{x} \in \mathbf{X} : g(\mathbf{x}) \in B_k\}, \tag{6.1.6}$$

is the domain in $\mathbf{X}$ that maps into the k^{th} bin B_k (see Fig. 6.1).

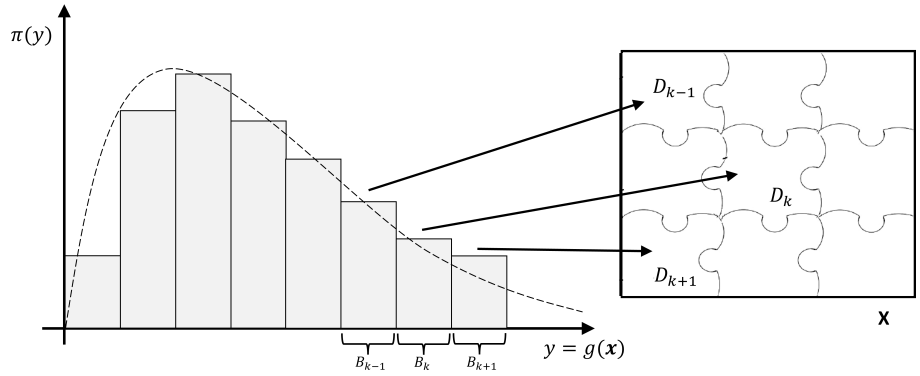


Figure 6.1: Schematic illustration of the connection between B_k and D_k.

While B_k are simple intervals, the domains D_k are multidimensional regions with possibly tortuous topologies. Therefore, define an indicator function to classify whether a given $\mathbf{x}$-value is in bin D_k or not:

$$I_{D_k}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in D_k; \\ 0, & \text{otherwise,} \end{cases} \tag{6.1.7}$$

or equivalently $\{y = g(\mathbf{x}) \in B_k\}$. By using this indicator function, the probability that y is in the k-th bin, i.e. $P_k = \mathbb{P}\{y \in B_k\}$, can be written as an integral in the input space:

$$P_k = \int_{D_k} p(\mathbf{x})d\mathbf{x} = \int I_{D_k}(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \mathbb{E}[I_{D_k}(\mathbf{x})]. \tag{6.1.8}$$

Where P_k can be estimated via a standard MC simulation. Namely, draw M i.i.d.

samples $\{\mathbf{x}_1, ..., \mathbf{x}_M\}$ from the distribution $p(\mathbf{x})$, and calculate the MC estimator of P_k using:

$$\widehat{P}_k^{MC} = \frac{1}{M}\sum_{j=1}^{M} I_{D_k}(\mathbf{x}_j) = \frac{M_k}{M}, \quad \text{for } k = 1, ..., K, \tag{6.1.9}$$

where M_k is the number of samples that fall in bin B_k. From $\{P_k\}_{k=1}^K$, the PDF of y at the point $y_k \in B_k$—for a sufficiently small Δ—can be calculated as $\pi(y_k) \approx P_k/\Delta$.

This approach can be readily adapted to reconstruct the distribution of $f(\mathbf{w}, \mathbf{Z})$ in the context of forecasted asset returns[3], from which one can obtain the expected return and conditional value-at-risk. Crucially, to obtain an accurate estimate of the expected return for a proposed allocation $\mathbf{w}$, one only needs a relatively small number of samples (from $\mathbf{Z}$). However, to obtain an accurate estimate of CVaR, one must use a very large number of MC samples (from $\mathbf{Z}$) to ensure that the tail distribution is properly assessed. For further details on Monte Carlo methods for CVaR estimation, refer to Hong et al. (2014).

6.1.4 Monte Carlo method conclusion

The primary advantage of the Monte Carlo method lies in its robustness; the accuracy of the Monte Carlo approximation is unaffected by the dimensionality d of the input space $\mathbb{R}^d$. Unlike certain other methods (e.g., numerical integration), the Monte Carlo method is not afflicted by the "curse of dimensionality" as long as samples can effectively be generated from the target distribution $p(\mathbf{x})$.

Relative error is a key measure of accuracy in the Monte Carlo method. For M samples and a target probability p, the relative error (RE) with a confidence interval of 95% is given by:

$$RE \approx 1/\sqrt{Mp}. \tag{6.1.10}$$

[3]Detailed later in this thesis.

As an example, for an event with a probability of order 1×10^{-9}, to obtain a relative error of less than 10%, a Monte Carlo simulation would need at least 10^{11} samples. If drawing and calculating each sample took one millisecond, the total computation time would be over 3 years! While the Monte Carlo method is effective for very simple models with high probability, it is prohibitively inefficient in estimating small probabilities. As such, importance sampling was developed to improve the computational efficiency of the Monte Carlo method by driving sampling to the regions of interest.

6.2 Importance sampling

6.2.1 General framework

Importance sampling (IS) was pioneered in the 1950s by Kahn and Marshall (1953). It enhances the accuracy of numerical estimates by artificially increasing the concentration of samples within the critical region, by drawing samples from a *proposal*, or *importance sampling* distribution, denoted as $q(\mathbf{x})$. This distribution is deliberately weighted to favour the region of interest, to skew sampling in the direction of interest. The estimates need to be adjusted to account for the fact that samples were drawn from the proposal distribution $q(\mathbf{x})$ rather than the original distribution $p(\mathbf{x})$.

Let $q(\mathbf{x})$ be an arbitrary distribution, then Eq. 6.1.1 can be rewritten as:

$$\mathbb{E}[y] = \int g(\mathbf{x})p(\mathbf{x})\ d\mathbf{x} = \int g(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\ d\mathbf{x} = \mathbb{E}_q\left[\frac{g(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})}\right], \tag{6.2.1}$$

where $\mathbb{E}_q[\cdot]$ denotes the expectation for $\mathbf{x} \sim q(\mathbf{x})$.

The ratio $p(\mathbf{x})/q(\mathbf{x}) = w(\mathbf{x})$ is called the *weight*, or *likelihood ratio*, and the inverse $B(\mathbf{x}) = w^{-1}(\mathbf{x})$ is called the *bias* at $\mathbf{x}$. The expectation can be simplified to:

$$\mathbb{E}[y] = \int g(\mathbf{x})w(\mathbf{x})q(\mathbf{x})\ d\mathbf{x} = \mathbb{E}_q[gw]. \tag{6.2.2}$$

Applying a Monte Carlo simulation to draw M samples $\mathbf{x}_1, ..., \mathbf{x}_M$ from the proposal distribution $q(\mathbf{x})$ and applying them to $\mathbb{E}_q[gw]$ obtains:

$$\widehat{g}_{IS} = \frac{1}{M}\sum_{j=1}^{M} g(\mathbf{x}_j)w(\mathbf{x}_j), \tag{6.2.3}$$

where $\widehat{g}_{IS}$ is the called the importance sampling estimator of $\mathbb{E}[g]$.

6.2.2 Importance sampling for failure probability

Importance sampling can be used to estimate the failure probability by drawing M samples $\mathbf{x}_1, ..., \mathbf{x}_M$ from the IS distribution $q(\mathbf{x})$ and then estimating P_F using the following equation:

$$P_F^{IS} \approx \frac{1}{M}\sum_{j=1}^{M} I_F(\mathbf{x}_j)w(\mathbf{x}_j), \tag{6.2.4}$$

where $w(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ and $I_F(\mathbf{x}_j)$ is an indicator function as defined in Eq. 6.1.4.

6.2.3 Importance sampling conclusion

Importance sampling allows for more efficient estimation of rare event probabilities and other risk measures. In the context of failure probability estimation, IS draw samples from a distribution that targets the failure regions, enabling precise estimation with significantly fewer samples than standard Monte Carlo simulations.

The performance of IS relies heavily on the choice of the IS or proposal distribution, which must be carefully selected to provide efficient sampling in the region of interest. IS can struggle with high-dimensional problems and those with multiple failure regions. To address these limitations, and further drive performance, more advanced methods have been developed—explored in the next chapter.

First, we present key methods for generating samples from target distributions. These sampling techniques are critical to the advanced methods discussed in the subsequent chapter. Understanding these techniques' strengths and weaknesses establishes a solid foundation for exploring the more sophisticated approaches to rare event simulation which follow—and it is necessary for the presentation of our proposed algorithm in Chapter 8.

6.3 Sampling methods

Rare event simulations—which can be used to determine the probability of a failure event—all rely on the ability to generate samples from target probability distributions, denoted by π. In this chapter, we present three popular and widely applicable sampling methods, namely Markov chain Monte Carlo (MCMC), sequential Monte Carlo sampler (SMCS) and population Monte Carlo (PMC).

6.3.1 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are pivotal tools in statistical inference, particularly for approximating complex probability distributions. These methods rely on constructing Markov chains whose stationary distribution aligns with the target posterior distribution. This section introduces the foundational concepts underpinning MCMC, starting with the basic principles of random walks, progressing to Markov chains, and then the Metropolis-Hastings algorithm.

Random walks

Random walks form the foundation for understanding Markov processes. A random walk is a stochastic process where an object's movement in a mathematical space is independent of its past positions.

Definition 6.3.1 (Random walks) *Let $X_1, X_2, \ldots$ be i.i.d. random variables, then*

the partial sum, $W_n = X_1 + X_2 + \ldots + X_n$, represents the position of the random walk at step n starting from $W_0 = 0$.

As $W_{n+1} = W_n + X_{n+1}$ for $n = 0, 1, 2, \ldots$, the random variables $W_1, W_2, \ldots$ are not independent, however, the difference $W_n - W_m$ for $m < n$ is independent of W_m.

Markov chains

The concept of Markov chains arises as a natural progression from random walks. While a random walk involves independent steps, a Markov chain introduces a dependency, albeit limited to the current state.

Definition 6.3.2 (Markov chains) *(Roberts and Rosenthal, 2004) A Markov chain is a stochastic process $\{X_n\}$ defined on a continuous state space $\mathscr{X}$, characterised by the Markov property: the future state X_{n+1} depends only on the current state X_n, not on the sequence of events that preceded it.*

Definition 6.3.3 (Transition kernel) *(Roberts and Rosenthal, 2004) The transition kernel $P(x, A)$ describes the probability of moving from a state x to a measurable set $A \subseteq \mathscr{X}$. It is defined as:*

$$P(x, A) = \mathbb{P}(X_{n+1} \in A \mid X_n = x),$$

where $P(x, A)$ is a probability measure for fixed $x \in \mathscr{X}$ and a measurable function of x for fixed A.

The n-step transition kernel $P^n(x, A)$ extends this definition to the probability of moving from state x to set A in exactly n steps, given by:

$$P^n(x, A) = \mathbb{P}(X_n \in A \mid X_0 = x),$$

where $X_0 = x$ is the starting state and X_n represents the state of the Markov chain after n transitions.

Stationary distributions

Markov chains extend the concept of random walks by introducing state dependency, forming the core concept behind MCMC methods. A stationary distribution π for a Markov chain is a probability measure that remains unchanged under the action of the transition kernel P. This means π satisfies:

$$\pi(A) = \int_{\mathscr{X}} P(x, A)\, \pi(dx) \quad \text{for all measurable } A \subseteq \mathscr{X}.$$

Definition 6.3.4 (Stationary Distribution) *(Roberts and Rosenthal, 2004) A probability measure π on $(\mathscr{X}, \mathcal{B})$ is a stationary distribution of the Markov chain with transition kernel P if $\pi P = \pi$, where $\pi P(A) = \int_{\mathscr{X}} P(x, A)\, \pi(dx)$ for all $A \in \mathcal{B}$.*

The Metropolis-Hastings algorithm

Metropolis and Ulam (1949) and Hastings (1970) proposed the Metropolis-Hastings algorithm as a statistical approach to solving differential equations. The algorithm is an MCMC method that constructs a Markov chain with a specified stationary distribution π, satisfying the detailed balance condition:

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx), \quad \forall x, y \in S$$

This condition is sufficient for ensuring that π is a stationary distribution of the Markov Chain with Kernel P. In the context of the Metropolis-Hastings algorithm, the condition is ensured by using an acceptance probability $a(x'|x)$ defined as follows:

$$a(x'|x) = \min\left\{1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right\},$$

where $q(x'|x)$ is the density of the proposal kernel, i.e.,

$$Q(x, dx') = q(x'|x)dx'.$$

This acceptance probability ensures that the ratio $\frac{\pi(x')P(x,dx')}{\pi(x)P(dx',x)}$ is equal to 1, thus satisfying the detailed balance condition. By adhering to this condition, the Metropolis-Hastings algorithm ensures that π is an invariant measure of the Markov chain.

Irreducibility and Aperiodicity

To ensure convergence of the Markov chain to the stationary distribution π, the chain must be irreducible and aperiodic. A Markov chain is said to be irreducible if it is possible to reach any measurable set with a positive measure from any state in a finite number of steps - formally defined as follows.

Definition 6.3.5 (Irreducibility) *(Roberts and Rosenthal, 2004) A Markov chain is irreducible with respect to a non-zero σ-finite measure ϕ on $\mathscr{X}$ if for any $x \in \mathscr{X}$ and any measurable set $A \subseteq \mathscr{X}$ with $\phi(A) > 0$, there exists an integer $n > 0$ such that $P^n(x, A) > 0$.*

A Markov chain is aperiodic if it does not cycle deterministically through states in fixed intervals - formally defined as follows.

Definition 6.3.6 (Aperiodicity) *(Roberts and Rosenthal, 2004) A Markov chain is aperiodic if it does not get trapped in a cyclic pattern, meaning there does not exist a fixed integer $d \geqslant 2$ and disjoint subsets $\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_d$ such that for all $x \in \mathscr{X}_i$, $P(x, \mathscr{X}_{i+1}) = 1$ and $P(x, \mathscr{X}_1) = 1$ for $x \in \mathscr{X}_d$.*

In the context of the Metropolis-Hastings algorithm, these properties guarantee that the Markov chain will eventually sample from the entire target distribution π

and facilitate the convergence of the chain to the desired posterior distribution, making the Metropolis-Hastings algorithm an effective tool for generating samples that approximate complex probability distributions. Conditions ensuring these properties for the Metropolis-Hastings algorithm are discussed in Roberts and Smith (1994) and Roberts and Rosenthal (2004).

Algorithm 6 Metropolis-Hastings algorithm

1: **Input**: Number of iterations T, initial value $\mathbf{x}_0$, target distribution $\pi(\mathbf{x})$, proposal distribution $q(\mathbf{x}'|\mathbf{x})$

2: **Initialization**

3: Let $t = 0$

4: **for** $t = 1$ to T **do**

5: **Draw a new candidate value**

6: Draw $\mathbf{x}'_t$ from the proposal distribution $q(\mathbf{x}'_t|\mathbf{x}_{t-1})$

7: **Calculate the MH acceptance probability**

8: Calculate $a(\mathbf{x}'_t|\mathbf{x}_{t-1}) = \min\left\{\frac{\pi(\mathbf{x}'_t)q(\mathbf{x}_{t-1}|\mathbf{x}'_t)}{\pi(\mathbf{x}_{t-1})q(\mathbf{x}'_t|\mathbf{x}_{t-1})}, 1\right\}$

9: Update $\mathbf{x}_t$ based on $a_t(\mathbf{x}'_t|\mathbf{x}_{t-1})$:

$$\mathbf{x}_t = \begin{cases} \mathbf{x}'_t, & \text{with probability } a(\mathbf{x}'_t|\mathbf{x}_{t-1}) \\ \mathbf{x}_{t-1}, & \text{otherwise} \end{cases} \tag{6.3.7}$$

10: **end for**

11: **Return**: Sampled values $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$

The Metropolis-Hastings algorithm is an effective sampling technique due to its simplicity and flexibility, particularly as the input PDF need only be known up to a multiplicative constant. Whilst a chain of samples generated by the algorithm will have a stationary distribution equal to the target distribution, it can be difficult

to know when the samples follow this distribution. Therefore, a burn-in period is often needed, where the first $P\%$ of samples are discarded; however, this reduces its efficiency.

A desirable feature for sampling methods is the generation of i.i.d. samples, this is not the case for MCMC. As such, autocorrelation adjustments may need to be applied to the generated samples, which can result in many samples being discarded. These limitations are addressed, at least in part, by the sequential Monte Carlo sampler (SMCS) and population Monte Carlo (PMC).

6.3.2 Sequential Monte Carlo sampler

Proposed by Del Moral et al. (2006), the sequential Monte Carlo sampler (SMCS) is a method for drawing samples from a sequence of distributions $\{q_t(\cdot)\}_{t=1}^{T}$. It is a generalisation of the particle filter for dynamic state estimation by Arulampalam et al. (2002), where weighted samples are generated from a posterior distribution. The SMCS procedure *moves* samples from a distribution $q_{t-1}(\cdot)$ to $q_t(\cdot)$ through:

1. **Forward kernel application** to each current sample, sometimes incorporating an acceptance criterion.

2. **Weight calculation** on each newly generated sample—detailed shortly.

3. **Effective sample size (ESS) check** across all samples. The ESS is calculated as follows:

$$\text{ESS} = \frac{\left(\sum_{j=1}^{M} w_j\right)^2}{\sum_{j=1}^{M} w_j^2}$$

   where w_j are the normalised weights of the samples. If the ESS falls below a specific threshold, usually less than half of the total number of samples, then:

4. **Resample** the proposed samples based on their weights. The resulting weighted samples follow the distribution $q_t(\cdot)$.

The paper by Beskos et al. (2014) provides a thorough theoretical analysis of SMCS, focusing on the algorithm's stability for high-dimensional problems. The paper demonstrates that SMCS is particularly powerful in sequential inference problems, approximate Bayesian computation and when there is a desire for a consistent adaptive algorithm with little user input.

The method

SMCS is central to our proposed algorithm in Chapter 8, so we now give a very detailed description and explanation of how SMCS can generate samples from a sequence of distributions $\{q_t(\cdot)\}$. We present the SMCS method in a recursive formulation, drawing on the work of Del Moral et al. (2006) and Wu et al. (2020).

Suppose that at time $t-1$, we have an IS distribution $\gamma_{t-1}(\mathbf{x}_{t-1})$, from which we can generate, or already have, an ensemble of M samples $\{\mathbf{x}_{t-1}\}_{j=1}^{M}$. To implement SMCS, we first define two conditional distributions $K_t(\cdot|\mathbf{x}_{t-1})$ and $L_{t-1}(\cdot|\mathbf{x}_t)$, referred to as the *forward* and *backward kernels* respectively. Using $L_{t-1}(\cdot|\mathbf{x}_t)$, we are able to construct a joint distribution of $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ in the form of

$$r_t(\mathbf{x}_{t-1}, \mathbf{x}_t) = q_t(\mathbf{x}_t) L_{t-1}(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{6.3.8}$$

such that the marginal distribution of $r_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ over $\mathbf{x}_{t-1}$ is $q_t(\mathbf{x}_t)$. Now, using $\gamma_{t-1}(\mathbf{x}_{t-1})$ and the forward kernel $K_t(\mathbf{x}_t|\mathbf{x}_{t-1})$, we can construct an IS distribution for $r_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ in the form of

$$\gamma(\mathbf{x}_{t-1}, \mathbf{x}_t) = \gamma_{t-1}(\mathbf{x}_{t-1}) K_t(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{6.3.9}$$

Using $\{\mathbf{x}_{t-1}\}_{j=1}^{M}$ and the forward kernel K_t, draw samples from this joint IS distribution $\gamma(\mathbf{x}_{t-1}, \mathbf{x}_t)$ to obtain an ensemble $\{(\mathbf{x}_{t-1}^j, \mathbf{x}_t^j)\}_{j=1}^{M}$. The corresponding weights

are computed as

$$\begin{aligned} w_t(\mathbf{x}_{t-1:t}) &= \frac{r_t(\mathbf{x}_{t-1}, \mathbf{x}_t)}{\gamma(\mathbf{x}_{t-1}, \mathbf{x}_t)} = \frac{q_t(\mathbf{x}_t)\ L_{t-1}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\gamma_{t-1}(\mathbf{x}_{t-1})\ K_t(\mathbf{x}_t|\mathbf{x}_{t-1})} \\ &= w_{t-1}(\mathbf{x}_{t-1})\, \alpha(\mathbf{x}_{t-1}, \mathbf{x}_t), \end{aligned} \tag{6.3.10a}$$

where

$$\begin{aligned} w_{t-1}(\mathbf{x}_{t-1}) &= \frac{q_{t-1}(\mathbf{x}_{t-1})}{\gamma_{t-1}(\mathbf{x}_{t-1})}, \text{ and} \\ \alpha_t(\mathbf{x}_{t-1}, \mathbf{x}_t) &= \frac{q_t(\mathbf{x}_t)\ L_{t-1}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q_{t-1}(\mathbf{x}_{t-1})\ K_t(\mathbf{x}_t|\mathbf{x}_{t-1})}. \end{aligned} \tag{6.3.10b}$$

As such the weighted ensemble $\{\mathbf{x}_{t-1:t}^j, w_t^j\}_{j=1}^N$ follows the joint distribution $r_t(\mathbf{x}_{t-1:t})$ and $\{\mathbf{x}_t^j, w_t^j\}_{j=1}^N$ follows the marginal distribution q_t. By repeating this procedure, one can obtain weighted samples from the sequence of distributions $\{q_t\}_{t=1}^T$.

SMCS implementation

For the SMCS method, the choice of forward and backward kernels is essential. While noting several methods exist for determining the optimal forward kernel, in our later application, we adopt the MCMC kernel proposed by Del Moral et al. (2006), which is closely related to the Metropolis step in MCMC (as the name suggests).

Specifically, the forward kernel (more precisely, the process for generating samples from the forward kernel) is constructed as follows. A proposal distribution $k(\mathbf{x}_t|\mathbf{x}_{t-1})$ is chosen and using a sample from the previous iteration $\mathbf{x}_{t-1}^j$, we draw a sample $\mathbf{x}_t^*$ from $k(\mathbf{x}_t|\mathbf{x}_{t-1}^j)$, and then accept (or reject) $\mathbf{x}_t^*$ according to the following acceptance probability:

$$a_t(\mathbf{x}_t^*|\mathbf{x}_{t-1}^j) = \min\left\{\frac{q_t(\mathbf{x}_t^*)}{q_t(\mathbf{x}_{t-1}^j)}\frac{k(\mathbf{x}_{t-1}^j|\mathbf{x}_t^*)}{k(\mathbf{x}_t^*|\mathbf{x}_{t-1}^j)}, 1\right\}. \tag{6.3.11}$$

That is, we set

$$\mathbf{x}_t^j = \begin{cases} \mathbf{x}_t^*, \text{ with probability } a_t(\mathbf{x}_t^*|\mathbf{x}_{t-1}^j) \\ \mathbf{x}_{t-1}^j, \text{ otherwise.} \end{cases} \tag{6.3.12}$$

Once a forward kernel $K_t(\mathbf{x}_t|\mathbf{x}_{t-1})$ is chosen, one can determine an optimal choice of L_{t-1} by:

$$\begin{aligned} L_{t-1}^{opt}(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \frac{q_{t-1}(\mathbf{x}_{t-1})K_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{q_t(\mathbf{x}_t)} \\ &= \frac{q_{t-1}(\mathbf{x}_{t-1})K_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{\int q_{t-1}(\mathbf{x}_{t-1})K_t(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_{t-1}}, \end{aligned} \tag{6.3.13}$$

where the optimality is achieved through yielding the minimal estimator variance (Del Moral et al., 2006). In reality, this optimal backward kernel usually cannot be used directly as the integral on the denominator cannot be calculated analytically. However, when the MCMC kernel is used, an approximate optimal kernel can be derived[4] from Eq. (6.3.13), as:

$$L_{t-1}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q_t(\mathbf{x}_{t-1})K_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{q_t(\mathbf{x}_t)}. \tag{6.3.14}$$

When Eq. (6.3.14) is used, the incremental weight function $\alpha_t(\mathbf{x}_{t-1}, \mathbf{x}_t)$ in Eq. 6.3.10b, reduces to the following:

$$\alpha_t(\mathbf{x}_{t-1}, \mathbf{x}_t) = \frac{q_t(\mathbf{x}_{t-1})}{q_{t-1}(\mathbf{x}_{t-1})}. \tag{6.3.15}$$

Note that, interestingly, when Eq. (6.3.14) is used, only the previous sample is used in the weight calculation. Later, in our proposed algorithm, we use the MCMC kernel and Eq. (6.3.14) as the forward and backward kernels, respectively.

To alleviate sample degeneracy, a key step in SMCS is the resampling of samples according to their associated weights. In SMCS, typically, resampling is conducted

[4]The detailed derivation can be found in Del Moral et al. (2006).

when the effective sample size (ESS) is lower than a prescribed threshold value $ESS_{\min}$ (Doucet and Johansen, 2009). To conclude, we provide the complete procedure of SMCS in Algorithm 7, to generate M samples from the target distribution $q_t(\cdot)$.

Algorithm 7 Sequential Monte Carlo sampler

Input: Number of iterations T, number of particles M, initial weighted ensemble $\{(\mathbf{x}_0^j, w_0^j)\}_{j=1}^M$, proposal function $k(\cdot|\cdot)$, target distribution $q(\cdot)$, $ESS_{\min}$
for $t = 1$ to T **do**
    for $j = 1$ to M **do**
        Draw $\mathbf{x}_t^*$ from $k(\cdot|\mathbf{x}_{t-1}^j)$
        Calculate the acceptance probability $a(\mathbf{x}_t^*, \mathbf{x}_{t-1}^j)$
        Determine $\mathbf{x}_t^j$ based on the acceptance probability
        Calculate α_t^j for weight adjustment
        Update weight: $w_t^j = w_{t-1}^j \cdot \alpha_t^j$
    end for
    Normalize the weights of the ensemble
    Calculate the Effective Sample Size (ESS)
    if $ESS < ESS_{\min}$ **then**
        Resample the ensemble
        Set $w_t^j = \frac{1}{M}$ for all $j = 1, \ldots, M$
    end if
end for
Return: Final ensemble $\{(\mathbf{x}_T^j, w_T^j)\}_{j=1}^M$

In the SMCS algorithm, each particle in the ensemble denoted as $\{(\mathbf{x}_t^j, \mathbf{w}_t^j)\}_{j=1}^M$, is updated independently, rendering the algorithm highly parallelisable. Specifically, the operations for drawing a new sample $\mathbf{x}_t^*$ from the proposal function $k(\cdot|\cdot)$, calculating the acceptance probability $a(\mathbf{x}_t^*, \mathbf{x}_{t-1}^j)$, updating the state $\mathbf{x}_t^j$, and adjusting weights α_t^j for each particle j can be executed simultaneously across multiple processors. This independent processing, devoid of sequential dependency between particles, significantly enhances computational efficiency, making SMCS particularly suited for distributed parallel computing environments. Moreover, the parallelisable nature of the resampling step, essential when the Effective Sample Size (ESS) falls below $ESS_{\min}$, further contributes to the scalability of SMCS, allowing for efficient handling of large-scale problems.

To summarise, the SMCS algorithm is easily parallelisable, which is the main advantage over MCMC; in addition, SMCS is designed for sampling from a sequence of target distributions—we will take advantage of both features in our later proposed algorithm.

SMCS implementation (kernel choice)

Critical to the success of SMCS is the kernel choice, Green et al. (2022) showed that the use of an approximately optimal L-kernel reduces the variance of SMCS estimates by up to 99% while also reducing the number of times that resampling was required by between 65% and 70%..

Recent developments in designing proposal or kernel distributions include Heng et al. (2020), which introduces an iterative scheme to approximate the solution of an associated optimal control problem for deriving proposal distributions. This scheme minimises the KL-divergence between the target and proposal distributions.

Another approach, proposed by Wu et al. (2020), involves using the Ensemble Kalman filters (EnKF) framework to construct kernels for SMCS. EnKF approx-

imates the posterior distribution with a Gaussian and derives the forward kernel from that approximation. Meanwhile, the backward kernel is based on a Gaussian approximation of the optimal backward kernel.

Finally, South et al. (2019) suggest using independent proposal distributions, which are highly parallelisable to exploit parallel computing power. This approach allows for estimating the posterior by reusing all MCMC candidates generated, resulting in significant ESS improvements.

SMCS implementation (resampling techniques)

Resampling is a crucial step in SMCS, where samples are selected based on their associated weights. Douc and Cappé (2005) provided a comprehensive overview of four major resampling techniques (Multinomial, Residual, Stratified, and Systematic) including numerical examples demonstrating the choice of resampling technique is highly problem-dependent. From a theoretical perspective, only residual and stratified resampling methods can be proven to outperform the basic multinomial resampling approach.

Further SMCS adaptations

Surrogate functions. Studies have explored the use of surrogate functions to improve the efficiency of Sequential Monte Carlo Sampling (SMCS) in dealing with sequences of distributions and overall performance functions. Recently, a paper by Bon et al. (2021) introduces a technique called delayed-acceptance (DA) to reduce the computational effort required to approximate the posterior expectation. This technique involves using a surrogate model to initially screen proposed samples and reject those that are likely to be rejected based on the surrogate model. This reduces the number of likelihood evaluations and computational expenses incurred by the full calculation with the actual likelihood function. The paper provides details on

designing the DA kernel and selecting appropriate parameters, which has demonstrated the efficiency of this method, particularly when dealing with computationally expensive performance or likelihood functions. The paper also addresses potential issues arising from mismatched tail probabilities in the target and surrogate models.

Nested SMCS. If SMCS faces challenges while sampling from complex and high-dimensional problems, Naesseth et al. (2015) proposed a possible solution known as "Nested SMCS". This approach employs a Nested Sequential Monte Carlo (NSMC) sampler to generate an efficient proposal distribution for the next NSMC sampler. The nesting can occur to any degree, making it a flexible solution. NSMC requires only approximate, properly weighted samples to be generated from the proposal distribution, making it usable within another NSMC procedure. The paper highlights the robust performance of this nested approach, particularly in handling difficult high-dimensional scenarios.

Summary of sequential Monte Carlo sampler

The sequential Monte Carlo sampler is a robust method for drawing samples from a sequence of target distributions, using weighted samples and resampling steps to ensure that samples effectively represent the evolving target distribution. The computational cost of SMCS can be high, especially when dealing with expensive likelihood evaluations, limiting its scalability for certain applications. Later in this chapter, we provide more details on the strengths and weaknesses of this method—but first, we introduce population Monte Carlo.

6.3.3 Population Monte Carlo

The population Monte Carlo method (PMC) was proposed in Cappé et al. (2004). In the context of PMC, the importance sampling distributions change in each iteration based on the performance of the previously generated samples. It has the advantage

of generating unbiased samples in every iteration. As a result, the process can be terminated at any point to produce a set of samples that are approximately simulated from the target distribution.

A useful reference text for PMC by Lee et al. (2011) focuses on how to adapt PMC for high-dimensional problems. In addition, the paper by Cappé et al. (2007) is useful for comparing sequential methods. We state the algorithm here and refer the reader to the paper by Cappé et al. (2004) for the complete derivation and rationale.

The algorithm

The population Monte Carlo Algorithm is as follows (for sample size M) as provided by Cappé et al. (2004).

Algorithm 8 Population Monte Carlo

1: **Input**: Number of iterations T, number of samples M, target distribution $\pi(\cdot)$
2: **for** $t = 1, ..., T$ **do**
3: **for** $j = 1, ..., M$ **do**
4: Select the proposal distribution $q_{jt}(\cdot)$
5: Generate $\mathbf{x}_j^{(t)} \sim q_{jt}(\cdot)$ and compute $w_j^{(t)} = \pi(\mathbf{x}_j^{(t)})/q_{jt}(\mathbf{x}_j^{(t)})$
6: Normalize the weights $w_j^{(t)}$
7: Resample N values from the $\mathbf{x}_j^{(t)}$'s using the normalized weights $w_j^{(t)}$
8: **end for**
9: **end for**
10: **Return**: Set of resampled values $\{\mathbf{x}_j^{(T)}\}_{j=1}^M$ and their weights $\{w_j^{(T)}\}_{j=1}^M$

PMC implementation

Key to the performance of PMC, in terms of sample quality and efficiency, is the choice of the proposal distribution $q_{it}(\cdot)$, which changes with the iteration index

t and sample index i. The most basic implementation is for the proposal distributions $q_{it}(\cdot)$ to be random walks centred at the previous samples $\mathbf{x}_i^{(t-1)}$. More complex implementations adapt to the quality of the samples from the previous step $(\mathbf{x}_1^{(t-1)}, ..., \mathbf{x}_N^{(t-1)})$, or, if storage allows, all previous samples generated across all iterations. In either case, a variance criterion is widely used to determine $q_{it}(\cdot)$.

PMC adaptation

Cross-entropy PMC. In Section 7.1, we will discuss the cross-entropy (CE) method, which is a popular technique for determining the optimal proposal distribution. Recently, Miller et al. (2021) proposed a cross-entropy based population Monte Carlo algorithm (CE-PMC).

The algorithm works by introducing a new parameter μ to parametrise the biasing distribution $q_{it}(\cdot)$. Samples are drawn from an adapted proposal distribution $\mathbf{x}_i^{(t)}$ from $q_{it}(\cdot, \mu_i^{(t)})$. The optimal value for $\mu_i^{(t)}$ is then determined using cross-entropy.

Numerical examples show that the proposed method is more computationally efficient than standard CE, especially for high-dimensional problems. The authors plan to investigate whether smoothing the performance functions could further enhance the algorithm's efficiency and performance.

Summary of population Monte Carlo

The main advantage of population Monte Carlo is its ability to generate unbiased samples at each iteration, enabling it to be used for adaptive importance sampling and effectively approximating target distributions. However, the performance of this technique is highly reliant on the selection of the proposal distributions, which can be challenging to design for complex, high-dimensional problems.

6.3.4 Comparison of sampling methods

Rare event simulations—as demonstrated in the next chapter—rely on the ability to sample from complex high-dimensional probability distributions effectively and efficiently. MCMC is the most widely used due to its efficiency, as the input PDF only needs to be known up to a multiplicative constant. However, there are drawbacks to MCMC, which SMCS and PMC address—as summarised in Table 6.3.4.

Table 6.1: Comparison of sampling methods

Aspect	**MCMC**	**SMCS**	**PMC**
Parallelism	Struggles with parallelism due to its serial nature. Cannot fully utilise advances in High-Powered computing.	Easily parallelisable, leveraging parallel HPC.	Easily parallelisable, leveraging parallel HPC.
Sample Correlation	Generates correlated samples, requiring autocorrelation adjustments and leading to inefficiencies.	Generates independent, uncorrelated samples.	Generates independent, uncorrelated samples.
PDF Error Estimation	Challenging to estimate the error in PDF estimation due to correlated samples (Lima et al., 2005).	Easier error estimation with independent samples.	Easier error estimation with independent samples.
Burn-in Period	Requires a significant burn-in period to reach stationary distribution.	Does not require a burn-in period.	Does not require a burn-in period.
Data Utilisation	Uses all data in a single batch, potentially missing out on sequential data structures (Gilks et al., 1995).	Updates posterior distribution with new data. Draws samples directly from the posterior.	Updates posterior distribution with new data. Draws samples directly from the posterior.

6.4 Conclusion

In conclusion, this chapter introduced rare event simulations to model events with minuscule probabilities. Such tail events pose unique challenges in risk analysis;

Monte Carlo methods offer a robust approach to tackle them, where MC's accuracy does not depend on the dimensionality of the input space, avoiding the "curse of dimensionality". However, MC's efficiency diminishes when estimating small probabilities due to the substantial number of samples required. Recognising this limitation, importance sampling techniques were established to enhance the accuracy and efficiency of MC simulations by using biasing distributions to generate more samples in the region of interest to quantify tail events better. More advanced techniques exist, which we explore next.

This chapter also introduced a key aspect of rare event simulation: the importance of generating samples from probability distributions. It highlighted three widely used sampling methods: Markov chain Monte Carlo (or Metropolis-Hastings) algorithm, sequential Monte Carlo sampler, and population Monte Carlo. These methods are fundamental to the more advanced algorithms for rare event simulation and failure probability estimation, which we explore in the next chapter—allowing us to continue towards our second objective: to develop new methodologies which better predict the probability and impact of tail events.

Chapter 7

Advanced Techniques for Risk Analysis

Quantifying uncertainty and tail risk is crucial for effective risk management. Monte Carlo and importance sampling methods can simulate rare events and estimate failure probabilities in simple problems. However, such approaches are inefficient for problems with high dimensionality, extremely small target probabilities, or complex multi-modal performance functions. Therefore, alternate numerical simulation techniques have been developed, including cross-entropy, subset simulation, line sampling, first-order reliability methods, and second-order reliability methods.

This chapter briefly introduces these approaches, drawing comparisons and highlighting limitations to motivate our proposed algorithm (presented in Chapter 8). Restating our problem set-up, let $\mathbf{x}$ be a random vector defined on $\mathbb{R}^d$ with probability density $p(\mathbf{x})$. Given a performance variable (y_g) defined by a function $y_g = g(\mathbf{x})$, the aim is to estimate the probability of a failure event, defined as:

$$F = \{\mathbf{x} \in \mathbb{X} |\ y_g = g(\mathbf{x}) > y_g^*\}. \tag{7.0.1}$$

The failure probability is defined as,

$$P_F = \mathbb{P}(F) = \int\limits_{\{\mathbf{x}\in\mathbb{X}|g(\mathbf{x})>y_g^*\}} p(\mathbf{x})d\mathbf{x} = \int\limits_{\mathbf{x}\in\mathbb{X}} I_F(\mathbf{x})p(\mathbf{x})d\mathbf{x}, \tag{7.0.2}$$

where $I_F(\mathbf{x})$ is an indicator function of the set F:

$$I_F(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in F; \\ 0, & \text{if } \mathbf{x} \notin F. \end{cases} \tag{7.0.3}$$

The failure probability can be estimated directly, or through determining the distribution $\pi(y_g)$ of the performance variable, from which the expectation of y_g and the failure probability of $y_g > y_g^*$ can be obtained.

7.1 Cross-entropy

Proposed by Rubinstein (1999), the cross-entropy (CE) method is one of the most widely used rare event simulations, which utilises Kullback-Leibler divergence to determine an optimal importance sampling distribution. We present CE based on De Boer et al. (2005).

7.1.1 The method

Kullback-Leibler (KL) divergence is a measure of divergence between two probability distributions, formally defined as:

Definition 7.1.1 *Let $p(\mathbf{x})$ and $q(\mathbf{x})$ be two continuous probability distributions. The Kullback-Leibler divergence $D_{KL}(p(\mathbf{x}), q(\mathbf{x}))$ is defined by*

$$D_{KL}(p(\mathbf{x}), q(\mathbf{x})) = \int_{-\infty}^{\infty} p(\mathbf{x})\ ln\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\ d\mathbf{x}. \tag{7.1.2}$$

CE tries to find an importance sampling, or biasing distribution $q(\mathbf{x})$ which is close to the target distribution $p(\mathbf{x})$, as measured by KL divergence[1]. Suppose $p(\mathbf{x})$ and $q(\mathbf{x})$ are continuous probabilities distributions, and that $q(\mathbf{x})$ is from a distribution family Q, we obtain the following optimisation problem:

$$\min_{q \in Q} D_{KL}(p(\mathbf{x}), q(\mathbf{x})) = \min_{q \in Q} \int_{-\infty}^{\infty} p(\mathbf{x}) \; ln \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) \; d\mathbf{x}. \tag{7.1.3}$$

Expanding the KL divergence obtains:

$$\begin{aligned} D_{KL}(p(\mathbf{x}), q(\mathbf{x})) &= \int_{-\infty}^{\infty} p(\mathbf{x}) \; ln \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) \; d\mathbf{x} \\ &= \int_{-\infty}^{\infty} p(\mathbf{x}) \; ln(p(\mathbf{x})) \; d\mathbf{x} - \int_{-\infty}^{\infty} p(\mathbf{x}) \; ln(q(\mathbf{x})) \; d\mathbf{x}. \end{aligned} \tag{7.1.4}$$

As the first term is a constant, the optimisation problem can be written as:

$$\max_{q \in Q} \int_{-\infty}^{\infty} p(\mathbf{x}) \; ln(q(\mathbf{x})) \; d\mathbf{x}, \tag{7.1.5}$$

equivalent to,

$$\max_{q \in Q} \int_{-\infty}^{\infty} \frac{g(\mathbf{x}) p(\mathbf{x})}{\mathbb{E}[g]} \; ln(q(\mathbf{x})) \; d\mathbf{x}, \tag{7.1.6}$$

or

$$\max_{q \in Q} \int_{-\infty}^{\infty} g(\mathbf{x}) p(\mathbf{x}) ln(q(\mathbf{x})) \; d\mathbf{x}, \tag{7.1.7}$$

where each term is known.

To make the optimisation problem easier to solve, one can consider $q(\mathbf{x})$ in a parametrised form $q(\mathbf{x}|\theta)$, for θ in state space Ω. From which, the complete cross-entropy method can be stated (Algorithm 9)—to estimate the expectation of y_g.

[1]Note: KL divergence cannot be considered a distance measure, as it is not symmetric. The KL divergence from $p(\mathbf{x})$ to $q(\mathbf{x})$ differs from $q(\mathbf{x})$ to $p(\mathbf{x})$.

Algorithm 9 Cross-entropy method

1: **Input**: Number of samples M, initial distribution $p(\mathbf{x})$, modified distribution $q(\mathbf{x}|\theta)$, function $g(\mathbf{x})$
2: Draw M samples $\mathbf{x}_1, ..., \mathbf{x}_M$ from $p(\mathbf{x})$
3: Solve the optimization problem to find θ:

$$\underset{\theta \in \Omega}{\operatorname{argmax}} \widehat{F}(\theta) = \frac{1}{M} \sum_{j=1}^{M} g(\mathbf{x}_j) \ln(q(\mathbf{x}_j|\theta))$$

4: Draw M samples $\mathbf{x}_1, ..., \mathbf{x}_M$ from $q(\mathbf{x}|\theta)$
5: Estimate $\widehat{g}_{IS} = \frac{1}{M} \sum_{j=1}^{M} g(\mathbf{x}_j) w(\mathbf{x}_j)$, where $w(\mathbf{x}_j) = \frac{p(\mathbf{x}_j)}{q(\mathbf{x}_j|\theta)}$
6: **Return**: Estimated value $\widehat{g}_{IS}$

To estimate failure probability using CE, replace the function $g(\mathbf{x})$ with an indicator function related to the failure event of interest—the remaining procedure is unchanged.

7.1.2 Cross-entropy summary

The cross-entropy method presents a robust framework for rare event simulation, leveraging the Kullback-Leibler divergence to optimise an importance sampling distribution. However, for very rare events, the method can struggle to find an appropriate biasing distribution.

To address this limitation, Li et al. (2011) propose a novel hybrid methodology that integrates surrogates with the cross-entropy (CE) method to enhance computational efficiency. This innovative technique diverges from the conventional CE optimisation procedure by operating on a surrogate, denoted as $\widehat{g}$, in lieu of the actual limit state function g. To bolster accuracy and circumvent potential loss when utilising surrogates, the study incorporates a hybrid strategy, building on earlier work by Li and Xiu (2010) to strategically integrate information from g when $\widehat{g}$ approaches zero. The results demonstrate clear and demonstrable improvements in efficiency compared to the standard CE implementation.

7.2 Subset simulation

Au and Beck (2001) introduced a novel approach for evaluating rare event failure probabilities called subset simulation (SS), which leverages intermediate failure events to streamline the calculation process. The paper by Au et al. (2007) validates SS's efficacy—through application to the Schueller tests[2]—underscoring SS's advancement in computational efficiency compared to traditional MC methods.

7.2.1 The method

In this section, we draw on the foundational work of Au and Beck, and insights from Zuev (2015). Subset simulation begins by decomposing the failure event F into a sequence of nested events F_k, such that

$$F = F_m \subset F_{m-1} \subset ... \subset F_1, \tag{7.2.1}$$

where F_1 is a common event (with high probability) and F_k are increasingly rare events, from $k = 1$ until $k = m$, the event of interest. Given this sequence, the probability of failure $\mathbb{P}(F)$ can be represented as the product of larger probabilities:

$$\begin{aligned} \mathbb{P}(F) &= \mathbb{P}(F_m) \\ &= \mathbb{P}(F_1)\frac{\mathbb{P}(F_2)}{\mathbb{P}(F_1)}\frac{\mathbb{P}(F_3)}{\mathbb{P}(F_2)}...\frac{\mathbb{P}(F_{m-1})}{\mathbb{P}(F_{m-2})}\frac{\mathbb{P}(F_m)}{\mathbb{P}(F_{m-1})} \\ &= \mathbb{P}(F_1) \cdot \mathbb{P}(F_2|F_1) \cdot ... \cdot \mathbb{P}(F_m|F_{m-1}), \end{aligned} \tag{7.2.2}$$

where $\mathbb{P}(F_k|F_{k-1}) = \mathbb{P}(F_k)/\mathbb{P}(F_{k-1})$ denotes the conditional probability of event F_k given the occurrence of event F_{k-1} for $k = 2, ..., m$.

In subset simulation, the conditional probabilities are determined adaptively as

[2]The Schueller tests evaluate and compare the effectiveness of numerical techniques in reliability estimation of structural systems in high-dimensional spaces.

the algorithm proceeds. A sampling method—like MCMC—is used to generate samples from each conditional distribution, from which each subset's probability can be estimated. After which, the target probability $\mathbb{P}(F_m)$ can be estimated using 7.2.2.

7.2.2 The algorithm

The best SS algorithm summary is in Zuev et al. (2012), which we adapt for our notation as Algorithm 10.

7.2.3 SS implementation

To ensure subset simulation is effective, it is crucial to choose an appropriate sampling method for transitioning from one set of samples to the next in each conditional probability stage. Typically, MCMC is used, where Zuev, in collaboration with Beck and Au, found the optimal acceptance/rejection level in MCMC should be around 23%, with a conditional failure probability of $p_0 = 0.2$ (Zuev et al., 2012).

If the step size within the sampling method is too large, there's a high likelihood of rejection during the acceptance/rejection step. On the other hand, if the step size is too small, the Markov Chain will explore the new distribution too slowly, requiring many samples; particularly in high-dimensional spaces. A simple SS adaptation to address this balancing dilemma is a component-wise Metropolis-Hastings method, which generates new samples by independently varying each dimension, thereby increasing the likelihood of sample acceptance—detailed in Papaioannou et al. (2015).

In SS, the intermediate failure region is determined so that the conditional failure probability at each level, $P(F_i|F_{i-1})$, equals a predefined value (e.g., $p_0 = 0.2$ as per Zuev et al.). However, achieving this exact probability at each level is not always feasible (Au et al., 2007)—a significant source of uncertainty in the final

failure probability estimation. To address this, Zuev et al. (2012) propose a subset simulation Plus method, which generates a posterior PDF of the failure probability, utilising both the sampled data and prior knowledge of the model—that is, a Bayesian approach.

Algorithm 10 Subset simulation

Input:

Set p_0, the conditional failure probability for $k = 0$;
Choose N, the number of samples per conditional level.

Algorithm:

Set $N_F(k) = 0$, number of failure samples at level k
Sample $\mathbf{x}_0^{(1)}, ..., \mathbf{x}_0^{(N)} \sim p(\mathbf{x})$
for $i = 1, ..., N$ **do**
    if $g^{(i)} = g(\mathbf{x}_0^{(i)}) > b$ **then**
        $N_F(k) \leftarrow N_F(k) + 1$
    end if
end for
while $N_F(k)/N < p_0$ **do**
    $k \leftarrow k + 1$
    Sort $\{g^{(i)}\} : \; g^{(i_1)} \leqslant g^{(i_2)} \leqslant ... \leqslant g^{(i_N)}$
    Define $b_j = \frac{g^{(i_{N-Np_0})} + g^{(i_{N-Np_0+1})}}{2}$
    for $j = 1, ..., Np_0$ **do**
        Starting from $\mathbf{x}_k^{(1),j} = \mathbf{x}_{k-1}^{(i_{N-Np_0+j})} \sim p(\cdot|F_k)$,
        generate $1/p_0$ states of $\mathbf{x}_k^{(1),j}, ..., \mathbf{x}_k^{(1/p_0),j} \sim p(\cdot|F_k)$, using MCMC
    end for
    Renumber: $\{\mathbf{x}_k^{(i),j}\}_{j=1,i=1}^{Np_0,1/p_0} \rightarrow \mathbf{x}_k^{(1)}, ..., \mathbf{x}_k^{(N)} \sim p(\cdot|F_k)$
    for $i = 1, ..., N$ **do**
        if $g^i = g(\mathbf{x}_k^{(i)}) > b$ **then**
            $N_F(k) \leftarrow N_F(k) + 1$
        end if
    end for
end while

Output:

$\widehat{p}_F^{SS}$, estimate of p_F:

$$\widehat{p}_F^{SS} = p_0^k \frac{N_F(k)}{N} \tag{7.2.3}$$

7.2.4 SS adaptations

This section briefly describes some popular SS adaptations and recent developments.

Hamiltonian Monte Carlo SS (HMC-SS) proposed by Wang et al. (2019) leverages Hamiltonian dynamics for sampling in the SS framework by transforming the probability space into Hamiltonian terms, treating the outcome event as a position vector and the probability structure as potential energy. HMC-SS, especially in its Rejection Sampling (RS-HMC) form, outperforms standard MCMC in exploring probability spaces for both Gaussian and non-Gaussian models. However, the Barrier Bouncing (BB-HMC) variant incurs higher computational costs, making it less suitable for general reliability problems.

Generalised SS (GSS) proposed by Cheng et al. (2022) modifies failure thresholds and amplifies input variables to more efficiently navigate towards the failure region. Utilising coordinate rotation and standard deviation amplification techniques similar to Single Value Decomposition, GSS has shown computational improvements over standard SS but is less effective for performance functions unrelated to the strongest response variation and in problems with more than a hundred dimensions.

Control variate SS (CV-SS) proposed by Abdollahi et al. (2020) is designed for problems with highly non-linear or misleading performance functions. CV-SS employs MCMC for sample generation, incorporating control variates for the proposal distribution and failure probability calculation. While promising, further analysis is required to evaluate its effectiveness against standard SS-MCMC and other methods.

Kriging methods proposed by Dubourg et al. (2011), are used to create surrogate models of complex performance functions, reducing computation time. It allows for error estimation within the surrogate models and final failure probability. However, this approach is less efficient with large design experiments and is unsuitable for non-smooth performance functions.

Splitting and hybrid SS was propsed by Au et al. (2007). Splitting SS is suitable for deterministic dynamical systems with stochastic excitations, while Hybrid SS is an extension that accommodates systems with uncertain parameters. These methods efficiently generate new samples but have limitations in thoroughly exploring the failure region and require specific system conditions for optimal performance.

Bayesian subset simulation (BSS) proposed by Bect et al. (2017) integrates subset simulation with Sequential Monte Carlo Sampling (SMCS)—as the sampling method within each conditional failure region iteration. BSS shows significant computational advantages over standard SS-MCMC. Future research aims to assess the method's robustness relative to the number of evaluations per intermediate level.

7.2.5 Subset simulation summary

Subset simulation is a sophisticated method for estimating rare failure probabilities, particularly in high-dimensional spaces. Its primary strength lies in its ability to efficiently compute these probabilities by decomposing the failure event into a series of nested events, thus simplifying the process and significantly improving computational efficiency compared to traditional Monte Carlo methods. However, SS is unable to reconstruct the entire distribution $g(\mathbf{x})$—rendering SS unsuitable for the estimation of VaR and CVaR—as these risk measures depend on the entire distribution of outcomes.

7.3 Further methods for failure probability estimation

We now provide a very brief summary of line sampling, the first-order reliability method (FORM), and the second-order reliability method (SORM).

7.3.1 Line sampling

Line sampling (LS) was devised for addressing high-dimensional challenges with non-linear performance functions and multiple failure regions (Koutsourelakis et al., 2004). LS utilises stepwise estimation, using information from the samples generated at each intermediate step, to probe the failure domain (similar to SS but with lines rather than points).

At the core of LS is the identification of an *important direction*, denoted as α, oriented towards the failure region(s). This involves utilising a sampling method, typically MCMC, to generate initial sample points, from which lines parallel to α are extended towards the failure regions. LS enables the estimation of failure probability by measuring the distance between each sample point and the limit state function.

The efficacy of LS was tested on the Schueller tests in a study by Pradlwarter et al. (2007). The tests revealed that LS is most effective in scenarios where the critical failure directions are identifiable and the system exhibits weakly non-linear reliability. The study concluded that in cases of increased complexity, universally applicable methods like subset simulation yield more reliable results.

Algorithm 11 Line sampling for failure probability estimation

1: **Input**: Target distribution $\pi(\cdot)$, failure domain F, number of samples M.
2: **Initialisation**: Estimate the most probable point (MPP) on the limit state surface.
3: **for** $j = 1$ to M **do**
4: Generate a random direction vector d_j from the MPP.
5: Perform a line search along d_j to find the intersection with the limit state function.
6: Sample points along the line segment within the failure domain F.
7: Estimate the failure probability contribution from this line.
8: **end for**
9: Aggregate the failure probability contributions from all lines.
10: **Return**: Estimated failure probability.

7.3.2 FORM and SORM

In structural reliability, the first-order reliability method (FORM) is widely used for assessing the failure probability. FORM transforms the problem into a standard normal space, where the most probable point (MPP) of failure is identified. The reliability index, β, is then computed, which is inversely related to the failure probability. FORM is particularly effective for linear and mildly non-linear problems, balancing accuracy and computational efficiency (Ditlevsen and Madsen, 1996; Rackwitz, 2001).

The second-order reliability method (SORM) extends FORM by including second-order terms in the Taylor series expansion of the limit state function around the MPP. This provides a more accurate failure probability estimation, especially in cases where the limit state surface is highly non-linear near the MPP (Hohenbichler and Rackwitz, 1987; Breitung, 1984). While SORM is computationally more intensive than FORM, its improved accuracy makes it valuable for complex engineering problems.

Algorithm 12 First-order reliability method (FORM)

1: **Input**: Limit state function $g(\mathbf{x})$, stochastic model of variables $\mathbf{x}$
2: **Initialisation**: Transform variables $\mathbf{x}$ to standard normal space $\mathbf{U}$
3: Find the design point $\mathbf{u}^*$ (most probable point of failure) in $\mathbf{U}$-space using an optimisation algorithm
4: Calculate the reliability index $\beta = -\Phi^{-1}(P_f)$, where P_f is the failure probability and Φ^{-1} is the inverse of the standard normal cumulative distribution function
5: **Return**: Reliability index β and failure probability $P_f = \Phi(-\beta)$

Algorithm 13 Second-order reliability method (SORM)

1: **Input**: Limit state function $g(\mathbf{x})$, stochastic model of variables $\mathbf{x}$
2: **Initialisation**: Transform variables $\mathbf{x}$ to standard normal space $\mathbf{U}$
3: Find the design point $\mathbf{u}^*$ (most probable point of failure) in $\mathbf{U}$-space using FORM
4: Approximate the limit state surface at $\mathbf{u}^*$ by a second-order (quadratic) surface
5: Calculate the curvature radii κ_i at $\mathbf{u}^*$
6: Use the curvature information to correct the FORM failure probability, adjusting for second-order effects
7: **Return**: Adjusted failure probability P_f accounting for second-order effects

7.3.3 Comparison

Objective two of this thesis ("Risk Management") is to develop methodologies that better predict the probability and impact of tail events in financial markets. This objective is motivated by the recognition that traditional financial models often fail to accurately predict and manage extreme market scenarios, such as "Black Swan" events. Additionally, this thesis aims to ensure uncertainty—from financial markets and mathematical models—is considered in investment risk management.

This chapter set out to provide a comprehensive evaluation of risk management strategies and rare event simulation methodologies. They have been discussed in the general setting—now we provide comments specifically concerning risk management in financial markets and portfolio optimisation. The paper by Schuëller et al. (2004) compares rare event simulation methods in the context of structural engineering, we draw on this review.

Direct MC is straightforward, it can handle multiple design criteria and points with high accuracy and is suitable for high-dimensional problems, but it is highly inefficient for estimating small probabilities due to its requirement for a large number of samples. This inefficiency is particularly problematic in financial markets, where rare events, though infrequent, can have substantial impacts—the inefficiency of MC in this context is emphasised in recent studies (see e.g., Embrechts et al. (2013)).

FORM is effective for linear limit state functions, offering high efficiency and suitability for medium-dimensional problems, but its accuracy is very low, especially for multiple design criteria. SORM incorporates second-order approximations to handle non-linearities. However, it has limited efficiency and accuracy, particularly for high-dimensional problems. This limitation becomes critical in complex financial models where non-linearity and high dimensionality are common. This criticism of FORM and SORM in handling complex, real-world scenarios is supported by Rackwitz (2001).

Importance sampling (IS) has improved efficiency compared to MC and is more robust than FORM and SORM. However, determining the optimal distribution for complex problems can be challenging. IS's performance in medium-dimensional problems is notable, but its application in more complex scenarios encountered in financial markets can be problematic, as noted in the works of Glasserman (2004).

Subset simulation (SS) breaks down the problem of failure estimation into intermediate stages to efficiently estimate small probabilities. It is widely applicable, performing well irrespective of the nature of the performance function and excelling in high dimensions. However, its tendency to generate biased samples and provide only a single probability value for a failure event limits its application in financial risk management, where a more comprehensive understanding of the risk distribution is crucial.

Finally, line sampling (LS) performs well in high-dimensional problems, identifying critical failure directions. Its efficiency, however, depends on an assumption of independence in sampling. The reliance on this assumption can be a drawback in financial markets modelling, where correlations between assets are common, as highlighted by McNeil et al. (2015).

Schuëller et al. (2004) produced a useful summary table for comparing rare event

simulation methods (albeit, for structural reliability problems)—it is reproduced here for reference.

Method	Multiple design criteria	Multiple design points	Accuracy	Dimension	Efficiency
Direct MC	Yes	Yes	High	High	Low
FORM	Yes	Yes	Very low	Medium	High
SORM	No	No	Low	Medium	Low
Standard IS	Yes	No	Medium	Medium	Medium
IS with kernel density estimator	Yes	Yes	High	Medium	Medium
SS	Yes	Yes	High	High	High
LS	Yes	Yes	High	High	High
LS using stepwise estimators	Yes	Yes	High	High	High

Table 7.1: Comparison of rare event simulation techniques (Schuëller et al., 2004)

7.4 Motivation

As discussed in Chapter 5, to capture tail risk and accommodate uncertainty, equity and credit risk management hinges on the ability to reconstruct probability distributions entirely. In equity risk, distribution reconstruction is pivotal for accurately evaluating the probability of extreme market movements. For credit risk, especially in copula models, the challenge extends to modelling joint default distributions—to assess the risk of clustered defaults. In order to fully assess the risks of potential failures, it is important to obtain the complete probability distribution of a performance variable to capture both the likelihood and severity of failures.

Obtaining the complete distribution of forecasted returns is necessary in portfolio optimisation too: from robust optimisation, which focuses on mean and variance (Du and Chen, 2004), to risk-based optimisation, where tail probability and extreme quantiles are of interest (Rockafellar and Uryasev, 2000); and finally, utility optimisation, where the entire distribution of the performance variable is required (Hazelrigg, 1998). In summary, all three areas – equity risk, credit risk, and failure probability estimation – necessitate the detailed reconstruction of probability distributions. However, the methods explored so far can only predict the probability of a single failure event. Our focus is on developing a new algorithm that can obtain the entire probability distribution of $g(\mathbf{x})$.

Chapter 8

The Multicanonical Sequential Monte Carlo Sampler

The inherent uncertainty within financial markets significantly influences investment performance and risk. These uncertainties stem from various sources, such as economic variables, market dynamics, investor behaviour, and external shocks. In risk management, a central task is to characterise and quantify these uncertainties.

Mathematical models and simulations are important tools to assess how systems are impacted by uncertainty. Within these, the *performance variable* y can be considered as a performance measure, expressed by a function $y = g(\mathbf{x})$, where the multi-dimensional random variable $\mathbf{x}$ represents all the uncertainty factors affecting the system; the performance function is usually not of analytical form and needs to be evaluated by simulating the underlying mathematical model. In finance, y could represent the future return distribution for a specific portfolio allocation.

Several advanced Monte Carlo techniques exist to provide a variance-reduced estimator for a quantity associated with the distribution of y, such as the failure probability. Our research has shown that to capture tail risk fully, it is necessary to obtain the complete probability distribution of $y = g(\mathbf{x})$.

One method for reconstructing probability distributions is 'multicanonical Monte Carlo' (MMC); a form of adaptive importance sampling. In MMC, the state space of the performance variable is split into a set of small bins, from which a (so-called) flat-histogram importance sampling distribution is constructed. This IS distribution ensures sampling occurs with equal probability for each bin; as such, MMC can obtain the entire distribution of y, including the tails, significantly more efficiently than using standard MC[1].

Brief history of MMC

In 1991 and 1992, Bernd Berg and Thomas Neuhaus published two papers presenting a new approach for simulating first-order phase transitions in Physical Systems called *multicanonical ensemble* (Berg and Neuhaus, 1991, 1992). Almost a decade later, a follow-up paper by Berg (2000) applied multicanonical Monte Carlo to simulating physical systems.

MMC was first applied to statistics by Yevick (2002), in the paper *'Multicanonical communication system modelling - application to PMD statistics'*. The paper used MMC to quantify the impact of polarization mode dispersion (PMD) on outages in optical fibre systems. As outage probability is typically of order 10^{-5} or less, standard MC would be too inefficient, so Yevick used MMC to generate the probability density function (PDF) of the outage probability under different scenarios. Subsequently, many papers applying MMC to topics in optimal communication have been published (see e.g., Biondini et al. (2002); Holzlööhner and Menyuk (2003); Bononi et al. (2009)).

[1]We note that an alternative approach would be to use surrogate models approximating the performance function; however, this is an entirely different strategy from developing sampling schemes. In certain applications, surrogate methods struggle (e.g. when the performance function is of high dimensions (Koziel et al., 2011), like in Bayesian ARMA-GARCH models), so sampling methods like MMC are more appropriate; therefore, this thesis restricts itself to such sampling methods.

MMC (primarily used in communications system engineering) has yet to be applied to financial modelling. With some adaptation and enhancements, we believe MMC can fill the gap in current risk methods, where complete probability distribution reconstruction is required. In the next few sections, we present the MMC algorithm, drawing on the presentations by Bononi et al. (2009) and Wu and Li (2016) and provide guidance on optimal implementation. This work builds to the presentation of our proposed algorithm, called 'multicanonical sequential Monte Carlo sampler'; an adaptation of MMC which allows for parallel implementation, more efficient sampling, and specific application to financial modelling.

8.1 Problem setup

Let $\mathbf{x}$ be a d-dimensional random vector following distribution $p(\mathbf{x})$ and let y be a scalar random variable characterised by a function $y = g(\mathbf{x})$. We want to determine the probability density function (PDF) of y, given by $\pi(y)$, where we assume that both $\mathbf{x}$ and y are continuous random variables.

In Section 6.1.3, we detailed how $\pi(y)$ could be estimated using a Monte Carlo simulation. In summary, the bounded support $R_y = [a, b]$ of $\pi(y)$ is decomposed into K bins of equal width Δ centred at the discrete values $\{b_1, ..., b_K\}$, implicitly defining a partition of the input space X into K domains $\{D_i\}_{k=1}^{K}$. We reprint the illustration of the connection between distribution $\pi(y)$ and the mapping B_k to D_k.

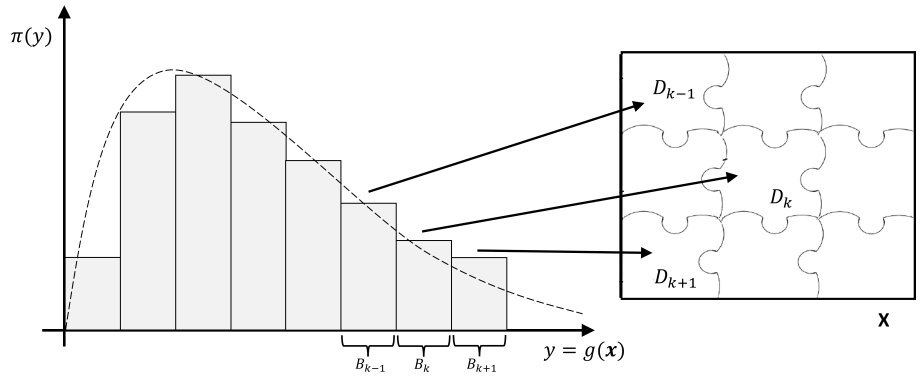


Figure 8.1: Schematic illustration of the connection between B_k and D_k.

By defining an indicator function $I_{D_k}(\mathbf{x})$, which classifies if an $\mathbf{x}$-value is in bin D_k or not, the probability that y is in the k^th bin, i.e. $P_k = \mathbb{P}\{y \in B_k\}$, can be written as an integral in the input space:

$$P_k = \int_{D_k} p(\mathbf{x})d\mathbf{x} = \int I_{D_k}(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \mathbb{E}[I_{D_k}(\mathbf{x})]. \tag{8.1.1}$$

Using this, P_k can be estimated via a standard MC simulation, by drawing M i.i.d. samples from $p(\mathbf{x})$ and then estimating P_k with:

$$\widehat{P}_k^{MC} = \frac{1}{M}\sum_{j=1}^{M} I_{D_k}(\mathbf{x}_j) = \frac{M_k}{M}, \quad \text{for } k = 1, ..., K, \tag{8.1.2}$$

where M_k is the number of samples that fall in bin B_k. From $\{P_k\}_{k=1}^K$, the PDF of y at the point $y_k \in B_k$ — for a sufficiently small Δ — can be calculated as $\pi(y_k) \approx P_k/\Delta$.

8.1.1 Flat histogram importance sampling

To accurately reconstruct $\pi(y)$, using a MC simulation, a large number of samples is required, particularly for narrow-tailed distributions. To improve efficiency, importance sampling can be utilised, to artificially increase the number of samples within

the histogram's tail bins. For IS distribution $q(\mathbf{x})$, Eq. 8.1.1 can be re-written as

$$P_k = \int I_{D_k}(\mathbf{x}) \left[\frac{p(\mathbf{x})}{q(\mathbf{x})}\right] q(\mathbf{x}) d\mathbf{x} = \mathbb{E}_q[I_{D_k}(\mathbf{x}) w(\mathbf{x})], \tag{8.1.3}$$

where $w(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ is the IS weight and $\mathbb{E}_q$ indicates expectation with respect to the IS distribution $q(\mathbf{x})$. The IS estimator for P_k can be written as:

$$\widehat{P}_k^{IS} = \left[\frac{1}{M} \sum_{j=1}^{M} I_{D_k}(\mathbf{x}_j) w(\mathbf{x}_j)\right], \tag{8.1.4}$$

for each bin $k = 1, ..., K$.

Within this framework, selecting the appropriate IS distribution $q(\mathbf{x})$ is particularly difficult as we are estimating multiple values (P_1, P_2, ..., P_K) instead of just one. A special IS distribution needs to be designed to:

1. Allocate the same probability to each bin, that is, assuming $\mathbf{x} \sim q(\mathbf{x})$, then

$$P_k^* := \mathbb{P}(y = g(\mathbf{x}) \in B_k) = 1/K,$$

   for all k. Intuitively, this property allows all bins to be equally visited by the samples generated from the IS distribution.

2. Assign a constant weight to all samples falling in the same bin, that is $w(\mathbf{x}) = \Theta_k$ for all $\mathbf{x} \in D_k$, where Θ_k is a positive constant. This ensures that all samples falling in the same bin are equally good.

To achieve this, the multicanonical Monte Carlo method proposes a so-called *uniform weight flat-histogram* (UW-FH) IS distribution to artificially elevate the probability of sampling points in less populated tail regions. The UW-FH distribution ensures that each bin has an equal probability of being sampled and that a constant weight

is applied within each bin, creating a more uniform and representative sampling of the entire distribution.

The UW-FH distribution can be defined in the form:

$$q(\mathbf{x}) \propto \begin{cases} \frac{p(\mathbf{x})}{c_\Theta \Theta(\mathbf{x})}, & \mathbf{x} \in D, \\ 0, & \mathbf{x} \notin D, \end{cases} \tag{8.1.5}$$

where $\Theta(\mathbf{x}) = \Theta_k$ for $\mathbf{x} \in D_k,\ k = 1, ..., K$, and c_Θ is a normalizing constant. Then the estimator for P_k becomes,

$$P_k^* = \int_{D_k} q(\mathbf{x}) d\mathbf{x} = \frac{\int_{D_k} p(\mathbf{x}) d\mathbf{x}}{c_\Theta \Theta_k} = \frac{P_k}{c_\Theta \Theta_k}. \tag{8.1.6}$$

By design $P_k^* = 1/K$ for all k, so $\Theta_k \propto P_k$, i.e. Θ_k is proportional to the sought probability P_k and $c_\Theta = \sum_{k=1}^K \frac{P_k}{\Theta_k}$.

8.1.2 Multicanonical Monte Carlo method

The UW-FH distribution, given by Eq. (8.1.5), cannot be used directly as Θ_k depends on the sought-after unknown P_k. The MMC method addresses this issue iteratively, starting from the original input PDF $p(\mathbf{x})$, to progressively refine the IS distribution, moving closer to the ideal UW-FH distribution. Starting with $q_0(\mathbf{x})$ and $\Theta_{0,k} = p$ for all $k = 1, ..., K$, where $p = \sum_{k=1}^K P_k$, the MMC method iteratively constructs a sequence of distributions (for $t \geqslant 1$),

$$q_t(\mathbf{x}) \propto \begin{cases} \frac{p(\mathbf{x})}{c_t \Theta_t(\mathbf{x})}, & \mathbf{x} \in D; \\ 0, & \mathbf{x} \notin D. \end{cases} \tag{8.1.7}$$

where $\Theta_t(\mathbf{x}) = \Theta_{t,k}$ for $\mathbf{x} \in D_k$ and c_t is the normalizing constant for q_t. Ideally, we want to construct q_t to converge to the actual UW-FH distribution as t increases,

so each iteration brings us closer to an equalised sampling across all bins, thereby achieving a more representative sampling of $\pi(y)$.

Accurately estimating $\{\Theta_{t,k}\}_{k=1}^{K}$ is crucial, as it directly influences the effectiveness in sampling the performance variable's distribution more evenly in each iteration. For IS distribution $q_t(\mathbf{x})$, we can see that $P_k = c_t P_k^* \Theta_{t,k}$. Therefore, in each MMC iteration, M samples $\{\mathbf{x}_j\}_{j=1}^{M}$ are drawn from the current IS distribution $q_t(\mathbf{x})$, then $\{\Theta_{t+1,k}\}_{k=1}^{K}$ is updated using the following formulas,

$$\widehat{H}_{t,k} = \frac{M_{t,k}^*}{M} \tag{8.1.8a}$$

$$P_{t,k} = \widehat{H}_{t,k}\ \Theta_{t,k} \tag{8.1.8b}$$

$$\Theta_{t+1,k} = P_{t,k} \tag{8.1.8c}$$

where $M_{t,k}^*$ is the number of samples falling into region D_k in the t^{th} iteration[2].

The process is repeated until the resulting histogram is sufficiently "flat" (see Section 8.2.2). A 'flat' histogram indicates that each bin has been sampled uniformly, signifying a successful implementation of the MMC method and an accurate representation of the full distribution.

To implement the MMC method, samples must be generated from the warped PDFs at each iteration. Typically, this is done using MCMC, which is particularly suited due to its ability to sample from complex distributions. Optimal MCMC acceptance probability ranges from $20-30\%$ for high-dimensional problems to $40-70\%$ for low-dimensional problems (Biondini et al., 2002).

The multicanonical Monte Carlo method using MCMC is given as Algorithm 14.

[2]In Eq. (8.1.8b), we neglect the normalizing constant c_t as it is not needed in the algorithm, which will become clear later.

Algorithm 14 Multicanonical Monte Carlo method—MCMC based

Input: Distribution $p(\mathbf{x})$, function $g(\mathbf{x})$, bins $\{B_k\}_{k=1}^{K}$, number of samples M

Initialisation: Set $q_0(\mathbf{x}) = p(\mathbf{x})$, $\Theta_{0,k} = p$ for all $k = 1, ..., K$

repeat

    Draw M samples $\{\mathbf{x}_j\}_{j=1}^{M}$ from the current IS distribution $q_t(\mathbf{x})$

    for $k = 1$ to K **do**

        Count $M_{t,k}^{*}$, the number of samples falling into region D_k

        Calculate $\widehat{H}_{t,k} = \frac{M_{t,k}^{*}}{M}$

        Update $P_{t,k} = \widehat{H}_{t,k} \cdot \Theta_{t,k}$

        Set $\Theta_{t+1,k} = P_{t,k}$

    end for

    Update $q_{t+1}(\mathbf{x})$ according to the new $\Theta_{t+1,k}$ values

until Histogram is sufficiently flat

Return: Estimates $\{\widehat{P}_k\}_{k=1}^{K}$ for the PDF $\pi(y)$ at points $y_k \in B_k$

8.2 MMC implementation

This section discusses key aspects of MMC implementation, including bin width selection, stopping criteria, and recent adaptations that enhance the algorithm's performance.

8.2.1 Bin width selection

A crucial aspect of MMC's performance is the choice of bin width. Bononi et al. (2009) suggests that the bin width should be set such that the probability in a bin is within one order of magnitude of its adjacent bins. This empirical approach ensures a balanced representation of the performance variable's distribution across all bins.

8.2.2 Stopping criteria

The algorithm is considered complete when the histogram is sufficiently "flat". Iba et al. (2014) propose when the number of samples in each bin is at least 92% of the average value across all bins, then it can be considered flat—as this is sufficient to achieve a comprehensive representation of $\pi(y)$.

8.2.3 MMC enhancements

To improve MMC's efficiency, several adaptations have been proposed, addressing specific challenges in its application:

Subset multicanonical Monte Carlo (SMMC): Chen and Li (2017) introduced SMMC, combining MMC with Subset Simulation for enhanced efficiency. SMMC focuses on generating samples from smaller, iteratively refined subdomains of the performance variable, moving closer to the region of interest with each iteration. This targeted approach improves algorithm performance by utilising more samples in estimating the density of the area of interest.

Surrogate accelerated MMC for UQ: Addressing the issue of computationally expensive performance functions, Wu and Li (2016) developed an adaptive algorithm using local Gaussian process (GP) surrogates. This method constructs surrogates within a non-parametric Bayesian regression framework, significantly speeding up each MMC iteration. The efficacy of the GP surrogate depends on the choice of covariance functions, highlighting a potential area for future research.

8.3 Adapting multicanonical Monte Carlo

In financial risk management, obtaining the probability distribution associated with a system's performance or reliability is crucial—as explained in Section 7.4. Despite its importance, research into efficient methods capable of reconstructing the

entire distribution of performance variables has been limited primarily due to the associated high computational costs. To address this directly, we propose several adaptations and enhancements to MMC, to improve efficiency and allow for parallel computing implementation. After which, we focus on the specific application of MMC to financial modelling.

8.3.1 New sampling approach

Each iteration of the MMC algorithm introduces a new importance sampling distribution, progressing towards the targeted UW-FH distribution. This process creates a sequence of distributions, each requiring efficient sampling.

It has been common practice to use Markov chain Monte Carlo (MCMC) to generate samples from the iteratively warped PDFs in each MMC iteration. However, MCMC has several limitations, explored in Section 6.3.4. Firstly, MCMC generates correlated samples, making it more difficult to estimate errors compared to independent sampling methods (Lima et al., 2005). Secondly, MCMC lacks preferential direction in exploring proposal distributions, leading to lower efficiency, as demonstrated by Bononi et al. (2009). Lastly, MCMC's focus on converging to a single stationary distribution can be a bottleneck, especially when the goal is to navigate through a series of evolving distributions.

In the context of MMC, an alternative yet underexplored method is the Sequential Monte Carlo sampler (SMCS). Specifically designed to generate samples from a sequence of distributions, SMCS aligns seamlessly with MMC's requirements. SMCS's inherent design makes it an apt choice for MMC, enabling a more efficient journey through the series of distributions; adapting to changes in the target space as the MMC algorithm evolves. Additionally, SMCS is easily implemented in a parallel computing environment, which is not true for MCMC.

Parallel computing

MCMC has an inherent serial nature and struggles with parallelisation. While parallel variants exist (see e.g., VanDerwerken and Schmidler (2013)), such methods do not fully leverage the capabilities of modern high-performance computing. Specifically, as MCMC requires a considerable 'burn-in' period to achieve convergence to the target distribution, even in a parallel setting, this is costly. As a result, MCMC is one of the most significant cost drivers in the MMC algorithm; we demonstrate this cost implication in our numerical examples.

In stark contrast, SMCS offers inherent parallelisability and does not require any 'burn-in' period, aligning well with contemporary computing architectures. This characteristic of SMCS is pivotal in enhancing the efficiency of the MMC method in financial applications, promising more accurate and comprehensive insights into financial risk management. The subsequent sections delve deeper into the implementation, discussing its advantages and exploring its potential applications in engineering problems and financial modelling.

8.4 The multicanonical sequential Monte Carlo sampler

This section introduces our proposed algorithm, the *multicanonical sequential Monte Carlo sampler* (MSMCS). This novel algorithm combines the strengths of multicanonical Monte Carlo with the Sequential Monte Carlo sampler. This hybrid approach is designed to tackle the challenges of effectively reconstructing the full probability distribution of performance variables. This task has been historically challenging in financial modelling, due to financial data's high-dimensional and complex nature.

The proposed MSMCS algorithm employs SMCS to generate samples in each iteration of the MMC method. SMCS's ability to handle a series of evolving distributions makes it an ideal fit for MMC's requirements, where each iteration introduces a new biasing distribution. Integrating SMCS as the sampling method requires two key adaptations.

Adaptation one: weighted samples

In the standard MMC method, samples generated using MCMC are unweighted. Therefore, the update procedure for Θ's—determined by the proportion of samples landing in each bin—is based on unweighted samples. When using SMCS, the generated samples are weighted. Therefore, we need to adjust the MMC update procedure for the Theta distributions. Specifically, we change how the value of $\widehat{H}_{t,k}$—the estimator of P_i—is determined. When using unweighted samples, the update procedure is determined by Eq. (8.1.8). When SMCS is used, the update procedure needs to be modified; Eq. (8.1.8a) becomes

$$\widehat{H}_{t,k} = \sum_{j=1}^{M} I_{D_k}(\mathbf{x}_j)\; w(\mathbf{x}_j). \tag{8.4.1}$$

This ensures that the unique characteristics of SMCS-generated samples are appropriately factored into the MMC framework.

Adaptation two: annealing

For SMCS to be effective, two successive distributions cannot be too far apart; otherwise, the samples will likely be rejected in the acceptance/rejection Metropolis step. Within the MMC method, there is no guarantee that the IS distributions obtained in two successive iterations are close to each other. For example, in the numerical experiments (which follow), we observed that for high-dimensional problems, such

an issue frequently appears in the first MMC step due to the difference in the initial distribution $q_0(\mathbf{x}) \sim p(\mathbf{x})$ and subsequent target distribution $q_1(\mathbf{x})$.

To address this issue, we include a simulated tempering process in the MSMCS method to smoothen the transition between distributions. We introduce a set of intermediate distributions between q_t and q_{t+1}, which SMCS is applied to. The difference in the IS distributions can be attributed to differences in the Θ-functions (i.e. $\Theta_t(\mathbf{x})$ and $\Theta_{t+1}(\mathbf{x})$), as per Eq. (8.1.7), therefore we choose a strictly increasing sequence of scalars $\{\alpha_s\}_{s=1}^S$ with $\alpha_0 = 0$ and $\alpha_S = 1$, such that the intermediate Θ-functions are:

$$\Theta_s(\mathbf{x}) = \alpha_s \Theta_{t+1}(\mathbf{x}) + (1 - \alpha_s)\Theta_t(\mathbf{x}). \tag{8.4.2}$$

It follows that the sequence of intermediate distributions $\{q_s\}_{s=0}^S$ can be defined accordingly via Eq. (8.1.7). Applying SMCS to this sequence of distributions, we ultimately yield samples from the target distribution $q_{t+1}(\mathbf{x})$.

When q_t and q_{t+1} are close to each other, SMCS can efficiently generate samples from q_{t+1} via the forward kernel using the samples from q_t, so the tempering process is not needed. For two consecutive IS distributions that are far apart, whilst introducing intermediate steps for generating samples from the next target distribution $q_{t+1}(\mathbf{x})$ increases the computational time, overall the MMC converges faster, offsetting this increased cost. In our proposed algorithm, tempering is only triggered when certain prescribed conditions are satisfied (e.g. $\|\Theta_t(\mathbf{x}) - \Theta_{t+1}(\mathbf{x})\|$ exceeds a threshold value).

Complete proposed MSMCS algorithm

Algorithm 15 Multicanonical sequential Monte Carlo sampler (MSMCS)

Input: Distribution $p(\mathbf{x})$, function $g(\mathbf{x})$, bins $\{B_k\}_{k=1}^{K}$, number of samples M, sequence $\{\alpha_s\}_{s=1}^{S}$

Initialisation: Set $q_0(\mathbf{x}) = p(\mathbf{x})$, $\Theta_{0,k} = p$ for all $k = 1, ..., K$

repeat

    Apply SMCS to generate samples from $q_t(\mathbf{x})$

    for $k = 1$ to K **do**

        Calculate $\widehat{H}_{t,k} = \sum_{j=1}^{M} I_{D_k}(\mathbf{x}_j) w(\mathbf{x}_j)$

        Update $P_{t,k} = \widehat{H}_{t,k} \cdot \Theta_{t,k}$

        Set $\Theta_{t+1,k} = P_{t,k}$

    end for

    if Tempering condition is met (e.g., $\|\Theta_t(\mathbf{x}) - \Theta_{t+1}(\mathbf{x})\|$ exceeds threshold) **then**

        Define intermediate $\Theta_s(\mathbf{x}) = \alpha_s \Theta_{t+1}(\mathbf{x}) + (1 - \alpha_s)\Theta_t(\mathbf{x})$

        Apply SMCS to the sequence of distributions $\{q_s\}_{s=0}^{S}$

    end if

    Update $q_{t+1}(\mathbf{x})$ according to the new $\Theta_{t+1,k}$ values

until Convergence criterion is met

Return: Estimates $\{\widehat{P}_k\}_{k=1}^{K}$ for the PDF $\pi(y)$ at points $y_k \in B_k$

8.4.1 Asympotitic properties

As discussed in the context of Bayesian evidence estimation, Berg and Neuhaus (1992) demonstrated that MMC is asymptotically exact, meaning that as the number of iterations increases, the estimated probability distribution converges to the true distribution. When MMC is combined with MCMC, the method remains asymp-

totically exact. This is primarily because MCMC itself is an asymptotically exact sampling method, meaning that given sufficient time, it will sample from the target distribution correctly. In the context of MMC-MCMC, this exactness is preserved through the iterative adjustment of the sampling distribution, which ensures that the entire state space is adequately explored, including the low-probability regions.

The MSMCS method, which integrates MMC with SMCS, also achieves asymptotic exactness. SMCS is known for its ability to generate independent and uncorrelated samples, which simplifies error estimation and enhances sampling efficiency. As with MMC-MCMC, the iterative nature of MMC ensures that as the number of iterations increases, the sampled distribution converges to the true distribution. The thesis highlights the superior performance of MSMCS over traditional MMC-MCMC, particularly in high-dimensional and complex problems, while maintaining the property of asymptotic exactness – demonstrated in the following numerical examples.

8.5 Numerical examples

Four numerical examples of increasing complexity[3] are provided to demonstrate the performance of the MSMCS algorithm, highlighting the significant time-saving advantages over traditional MMC-MCMC methods, especially when implemented in a parallel computing environment. Each numerical example demonstrates a different aspect of our proposed algorithm. After which, we apply MSMCS to credit and equity risk.

[3]By complexity, we are referring to the problem's dimensionality and the performance variable's rarity.

8.5.1 Problem 1: Chi-Square distribution

Introduction

Our first numerical demonstrates the efficacy of MMC, using both the SMCS and MCMC, in accurately reconstructing the Chi-square distribution—a widely recognised continuous distribution in statistical analysis, describing the sum of squared independent Gaussian random variables.

Problem set-up

By summing the squares of k independent Gaussian random variables with zero mean and unit variance, represented as:

$$y = \sum_{i=1}^{k} x_i^2, \tag{8.5.1}$$

the dependent variable y follows the Chi-square distribution with k degrees of freedom ($y \sim \chi^2(k)$). The analytical form of this PDF is known, providing a benchmark for assessing the reconstruction accuracy of the MMC methods.

Implementation details

We focus on a Chi-square distribution with $k = 20$ degrees of freedom. The implementation of both the MMC-MCMC and MSMCS methods had 20 iterations, each comprising 5×10^3 samples. Using the same setup for both methods allows for a direct and fair comparison of their performance. In the MMC-MCMC iterations, a single long chain of 5×10^3 samples was utilised without a burn-in period.

Results

The results are illustrated in Figure 8.2, presented on both linear and logarithmic scales. The figure compares the reconstructed Chi-square distributions from

MSMCS and MMC-MCMC against the known analytical solution. It also shows the absolute and relative errors compared to the true analytical solution.

Interpretation of results

Both methods closely approximate the exact analytical solution across the distribution, with the linear scale showing minimal absolute and relative errors. The logarithmic scale emphasises the precision of both methods in the tails, capturing the extremes. This numerical example highlights the accuracy and effectiveness of both MSMCS and MMC-MCMC in modelling the distribution, including the low-probability tail regions, with a relatively small total sample size.

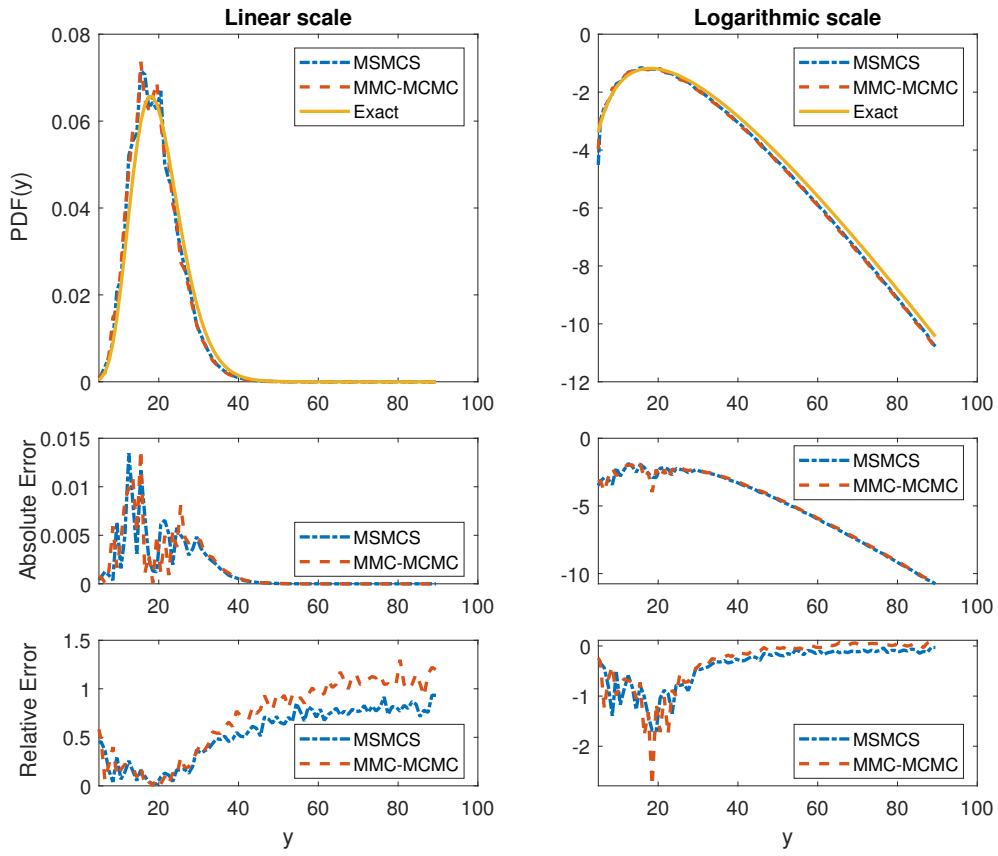


Figure 8.2: Chi-square distribution with 20 degrees of freedom computed by MSMCS and MMC-MCMC, compared to the analytical solution. The results are plotted on the linear scale (left column) and the logarithmic scale (right column). The first row contains the approximated and analytical PDFs of y. The second and third rows show the absolute and relative errors compared to the analytical solution.

8.5.2 Problem 2: Cantilever beam problem

Introduction

In the second numerical example, we explore a real-world engineering problem - the reliability analysis of a cantilever beam studied by Li et al. (2011) and Wu et al. (1990). In this example, we impose a burn-in period on MCMC, as is often required, to ensure all the samples generated by MCMC follow the MMC distribution in each

iteration—this is not required for SMCS, where all samples can be utilised.

Problem set-up

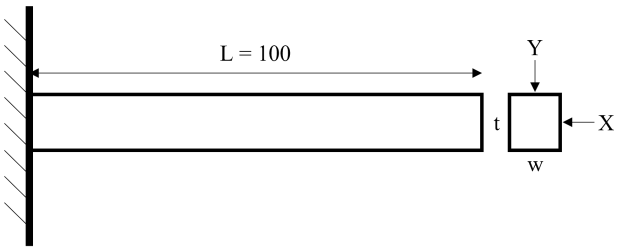


Figure 8.3: Cantilever beam

The cantilever beam (shown in Figure 8.3) is characterised by its width w, height t, length L, and elasticity E. Subjected to transverse load Y and horizontal load X, its failure is associated with the maximum deflection y, calculated using the equation:

$$y = \frac{4L^3}{Ewt}\sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2} \tag{8.5.2}$$

Following the problem set up of Li et al. (2011) and Wu et al. (1990), the beam is chosen to have fixed length $L = 100$, with w, t, X, Y and E treated as independent random variables following a Normal distribution, with means and variances detailed in Table 8.1.

Parameter	w	t	X	Y	E
Mean	4	4	500	1000	2.9×10^6
Variance	0.001	0.0001	100	100	1.45×10^6

Table 8.1: The mean and variance of the random parameters

Implementation details

The PDF of y is computed with three methods: plain MC, MMC-MCMC and MSMCS. In the MC simulation, 10^8 full model evaluations are conducted. Both

MMC-MCMC and MSMCS employ 20 iterations with 5×10^4 samples per iteration, ensuring a fair comparison. MMC-MCMC has a 15% burn-in period—to ensure the samples follow the target distributions—a requirement not imposed on MSMCS. The range $R_y = [5.35, 6.80]$ is divided into 145 bins of width 0.01.

Results

Figure 8.4 depicts the computed PDFs using all three methods. The results are presented on linear and logarithmic scales to provide a comprehensive view of the performance in various probability regions.

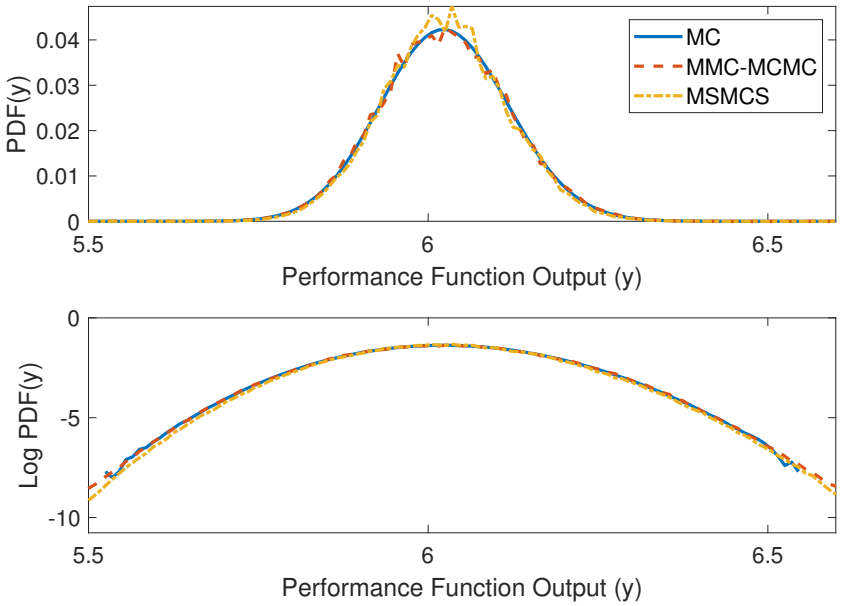


Figure 8.4: Cantilever beam PDF computed by MC, MSMCS and MMC-MCMC. The results are shown on both the linear scale (top) and the logarithmic scale (bottom).

Interpretation of results

While all three methods accurately reproduce the PDF in high-probability regions, the MMC-based methods (MMC-MCMC and MSMCS) exhibit superior performance in low-probability regions. Notably, plain MC fails to achieve the same

rarity level (e.g. at $y = 6.6$) with 100 times more samples. The two MMC methods yield comparable results in this example, but MSMCS has the advantage of parallel implementation.

8.5.3 Problem 3: Metaball limit-state function

Introduction

The numerical examples so far have simple performance functions; our third numerical example is specifically chosen to have a complex and changing topological structure. It involves analysing a metaball limit-state function, offering a unique perspective on the complexities encountered in probabilistic modelling. We also measure the time saved through the parallel implementation of MSMCS.

Problem set-up

The limit-state function under consideration is defined as (Breitung, 2019):

$$\begin{aligned} g(\mathbf{x}) = & \frac{30}{[4(x_1+2)^2/9 + x_2^2/25]^2 + 1} \\ & + \frac{20}{[(x_1-2.5)^2/4 + (x_2-0.5)^2/25]^2 + 1} - 5, \end{aligned} \tag{8.5.3}$$

where x_1 and x_2 are i.i.d. random variables following a standard Gaussian distribution. The function's geometry, featuring multiple regions of high probability, poses a significant challenge to many sampling methods, as studied in the paper by Tabandeh et al. (2022).

Implementation details

We compute the PDF of $g(\mathbf{x})$ with three methods: plain MC, MMC-MCMC and MSMCS. For the MMC-MCMC and MSMCS methods, we use 5 iterations with $15,000$ samples per iteration, and we compare these to a plain MC simulation comprising $75,000$ samples.

Results

The outcomes of the three methods are depicted in Figure 8.5. This figure presents the approximated PDFs on both linear and logarithmic scales.

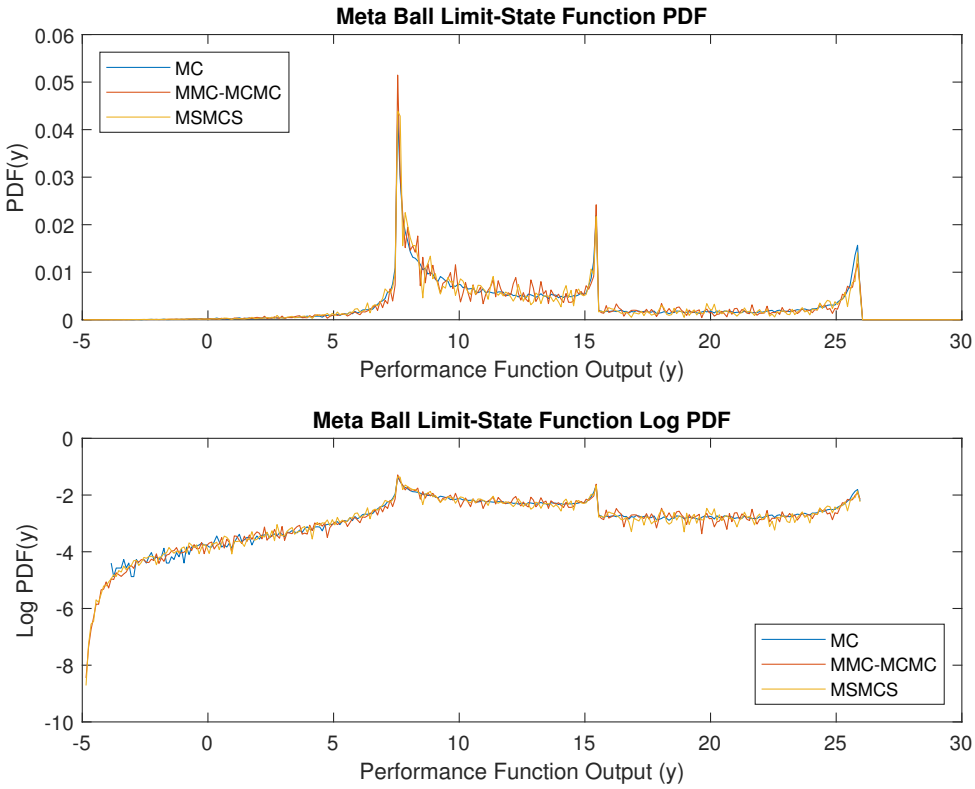


Figure 8.5: PDF computed by MC, MSMCS and MMC-MCMC. The results are shown on the linear scale (top) and the logarithmic scale (bottom).

Interpretation of results

All three methods yield similar performance in high-probability regions. However, a notable distinction is observed in the low-probability regions, where only the MMC methods effectively reproduce the sought PDF.

MSMCS's advantage in computational efficiency is demonstrated through leveraging parallel computing across multiple cores, significantly reducing computation time. To demonstrate the computational time saved, we provide the computation

time for the MSMCS algorithm across varying numbers of cores (using the previously detailed setup). The results are shown in Fig. 8.6.

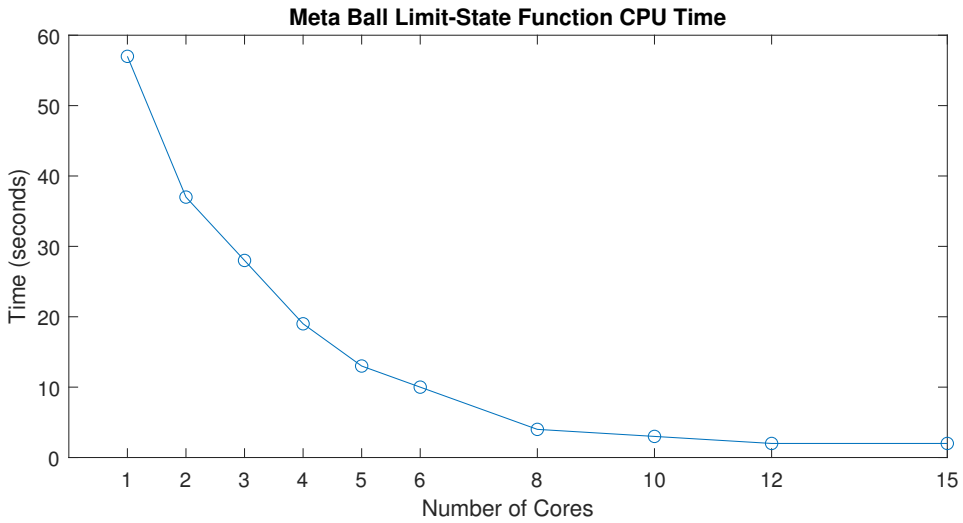


Figure 8.6: Time, in seconds, to complete 5 iterations of the MSMCS algorithm with 15000 samples per iteration and varying numbers of cores.

There is a minimum fixed cost in the MSMCS algorithm relating to the MMC component. Through using parallel computers across multiple cores, significant time savings can be made in the SMCS procedure—reducing the overall computational cost for the SMCS algorithm.

In contrast, MMC-MCMC cannot be implemented in parallel; its computational time is similar to that of MSMCS with one core (namely, c. 60 seconds), demonstrating the significant computational savings achieved through implementing the proposed MSMCS algorithm in a parallel computing environment.

8.5.4 Problem 4: Quarter car model

Introduction

A quarter-car model is a classic representation of vehicle suspension systems, providing insights into the system's response to varying road surfaces. In this example, we

implement MMC-MCMC in two alternate ways to demonstrate the computational efficiency gained by using MSMCS over MMC-MCMC. This problem has very high dimensionality and a complex performance function.

Problem set-up

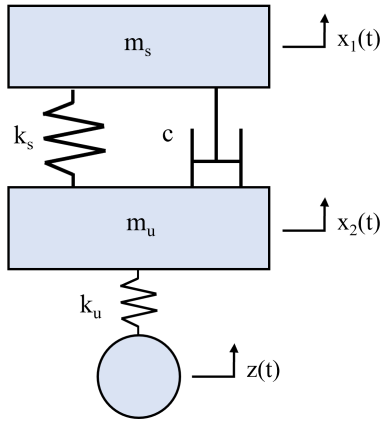


Figure 8.7: Quarter car model

We follow the problem set-up by Wong (2008). As illustrated in Figure 8.7, the Quarter Car model consists of a sprung mass m_s and an unsprung mass m_u connected by a non-linear spring (with stiffness k_s) and a linear damper (with damping coefficient c). The unsprung mass interacts with the road surface via the non-linear spring (with stiffness k_u). The displacement of the wheel $z(t)$ represents the interaction of the quarter-car system with the road surface. The parameters are assumed to be fixed, taking the values given by Table 8.2.

m_s	m_u	k_s	k_u	c
20	40	400	2000	600

Table 8.2: The parameter values of the quarter car model

The maximum displacement is the difference between the sprung and unsprung masses over a specific interval, as excessive displacement could indicate suspension

failure. The displacements of the sprung and the unsprung masses are denoted by x_1 and x_2, respectively. Mathematically, the model is described by a two-degree-of-freedom ordinary differential equation (ODE) system:

$$m_s \frac{d^2 x_1}{dt^2} = -k_s(x_1 - x_2)^3 - c\left(\frac{dx_1}{dt} - \frac{dx_2}{dt}\right), \tag{8.5.4a}$$

$$m_u \frac{d^2 x_2}{dt^2} = k_s(x_1 - x_2)^3 + c\left(\frac{dx_1}{dt} - \frac{dx_2}{dt}\right) + k_u(z(t) - x_2). \tag{8.5.4b}$$

The uncertainty arises through the random road profile $z(t)$, which is modelled as a zero-mean white Gaussian random force with standard deviation $\sigma = 1$. We are interested in the maximum difference between the displacements of the sprung and unsprung springs in a given interval $[0, T]$, as calculated by:

$$y = \max_{0 \leqslant t \leqslant T} \{|x_1(t) - x_2(t)|\}. \tag{8.5.5}$$

In extreme scenarios, the car's suspension would break when this displacement exceeds a certain value, say y^*. We want to reconstruct the entire probability density function (PDF) of y to estimate the probability $\mathbb{P}(y > y^*)$ for any value of y^*.

Implementation details

We numerically solve Eqs. (8.5.4) using the 4^{th} order Runge–Kutta method where the step size is taken to be $\Delta t = T/100$, effectively making it a 100-dimensional problem. Let $T = 1$ and set initial conditions of Eqs. 8.5.4 to be

$$x_1(0) = \frac{dx_1}{dt}(0) = 0, \quad x_2(0) = \frac{dx_2}{dt}(0) = 0. \tag{8.5.6}$$

In our simulation, the standard MC simulation is implemented with 10^6 samples. In both MSMCS and MMC-MCMC, 20 iterations with 2×10^4 samples per iteration are utilised.

As we explored in the previous example, the MSMCS method is easily parallelisable. Within each MMC iteration, SMCS can generate the new samples in parallel according to the target MMC distribution rather than forming a single long chain (as is required in MCMC). To ensure a fair computational comparison, we perform MMC-MCMC in two different ways. In the first implementation, we use a single long chain of length 2×10^4—the most typical implementation of MCMC, in line with the other numerical examples. In the second implementation, within each iteration, we use 10 chains of length 2×10^3 to provide a fairer comparison to the parallel implementation of MSMCS.

Results

The results of all three methods are shown in Figure 8.8. This include MC; MSMCS; MMC-MCMC with a single chain (referred to as MMC-MCMC-SC); and MMC-MCMC with multiple chains (referred to as MMC-MCMC-MC).

Interpretation of results

The simulation results, shown in a figure, reveal that while the standard Monte Carlo method provides a basic estimation of the PDF to the order of 10^{-6} (as expected), MSMCS achieves a far more detailed estimation, extending to much lower probability orders (10^{-12}). Comparatively, MMC-MCMC-SC with a single long chain accurately reconstructs the PDF. The parallel implementation of MMC-MCMC-MC with multiple short chains falls short, particularly in estimating higher displacement values ($y > 1.8$).

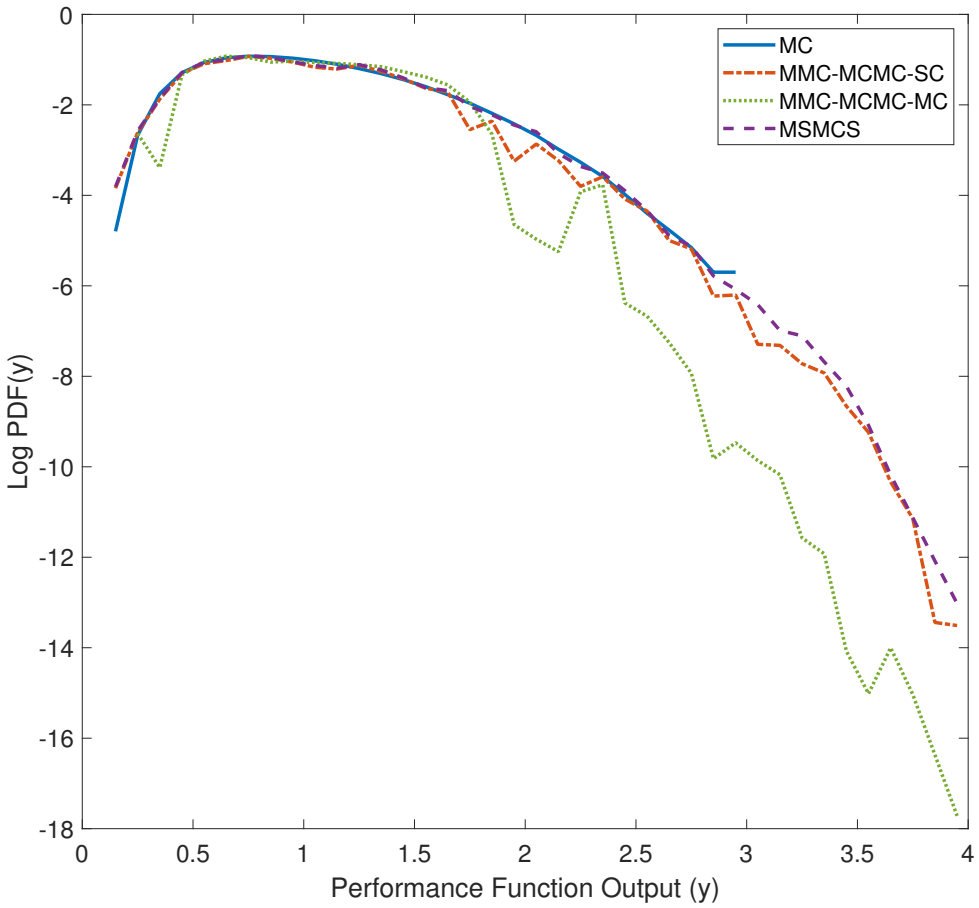


Figure 8.8: Quarter car model PDF computed by MC, MSMCS and MMC-MCMC. MMC-MCMC-SC uses a single long chain. MMC-MCMC-MC uses ten shorter chains in parallel. The results are shown on the logarithmic scale.

8.5.5 Summary of numerical results

The numerical examples offer compelling evidence of the proposed multicanonical sequential Monte Carlo sampler's capabilities in accurately and efficiently reconstructing probability distributions. These examples span simple statistical distributions to intricate real-world engineering models, each underscoring a different facet

of MSMCS's strengths.

The results highlight a critical limitation of MMC-MCMC when implemented in parallel using multiple short chains: its inability to accurately estimate the tail of the distribution. This contrasts sharply with the MSMCS method, which, due to its parallelisable design, not only provides more accurate results but does so with greater computational efficiency. The examples demonstrate MSMCS's superior performance over traditional MMC-MCMC methods in high-dimensional, real-world applications, particularly when leveraging parallel computing resources.

One weakness of the proposed method is that MCMC is easier to implement than SMCS and involves simpler computations—so MMC-MCMC is marginally faster than MSMCS. However, if one can use a parallel implementation, MSMCS significantly outperforms MMC-MCMC, as shown in the numerical examples. More importantly, both approaches to MMC can struggle in high-dimensional settings, where the generation of a new sample is likely to get rejected, which should be dealt with by developing and utilising more effective proposal distributions, for example, that based on the Hamiltonian dynamics (Neal, 2011).

8.6 Assessing credit risk using MSMCS

As detailed in Section 5.3, copula models play a pivotal role in credit risk modelling by enabling the separation of a portfolio's dependence structure from the marginal densities of individual risk factors. This separation allows for the individual risk of each obligor and the interconnectedness of different assets to be considered. The Normal copula model has been historically popular; however, it has faced significant scrutiny after its role in exacerbating the Global Financial Crisis came to light. The Student's t copula model has emerged as a more robust alternative. The heavier tails of the Student's t distribution allow the model to more accurately represent

the extremal dependence and the likelihood of simultaneous defaults, thus offering a more realistic measure of potential losses in a credit portfolio. The Student's t copula model, with its enhanced ability to capture tail risk, aligns better with the complex realities of financial markets, particularly in stress scenarios.

Current approach

It is possible to simulate a copula model using a Monte Carlo simulation. However, MC simulations are highly inefficient for high-dimensional problems or reconstructing distributions with low tail probabilities. For the Normal copula model, Glasserman and Li (2005) developed an importance sampling method involving a mean shift in the distributions of common risk factors Z and exponential twisting of the conditional default probabilities. This method significantly reduced variance and improved efficiency with asymptotic optimality.

For the Student's t copula models, Bassamboo et al. (2008) developed two importance sampling (IS) techniques, both targeting the denominator T's distribution. The first approach involves an exponential distribution twist, while the second employs a variation of hazard-rate twisting, referred to as ECM and HRT, respectively. Their study, which limits the copula model to just one common risk factor, presents a thorough theoretical and empirical analysis of the proposals. In the simulations, these methods demonstrate significant variance reductions.

Chan and Kroese (2010) offer two algorithmic variations on Conditional Monte Carlo (CMC). The first, CondMC, involves pre-selecting the proposal distribution, while the second, CondMC-CE, integrates cross-entropy (CE) at each algorithmic step. The paper's numerical examples, modelled after Bassamboo's, yielded variance reductions ranging between $1.2 - 4$ times and $20 - 100$ times higher than the ECM method by Bassamboo, using CondMC and CondMC-CE respectively. The CMC approach requires sampling from standard distributions, enhancing its efficiency

compared to the ECM method, which relies on rejection sampling and is typically three times slower than basic simulations.

Hong et al. (2014) strongly critiqued the proposed Conditional MC methods, highlighting a significant limitation: their performance heavily depends on the choice of conditioning variables, which vary per problem. This dependency poses a challenge in optimally selecting these variables, which must be tailored for each problem scenario.

Motivation for MSMCS application

There is still room for improvement in copula simulations, and this is where our algorithm, the multicanonical sequential Monte Carlo sampler, comes into play. MSMCS offers a novel approach to credit risk modelling, particularly for Student's t copula models, with four key advantages:

- **Efficient sampling of tail probabilities:** MSMCS is uniquely equipped to handle the challenges of sampling from distributions with low tail probabilities, a critical aspect in accurately assessing credit risk. This is especially important in Student's t copula models, known for their heavier tails and more complex dependence structures.

- **Adaptive importance sampling:** Unlike traditional importance sampling methods, which require careful selection of proposal distributions, MSMCS adaptively constructs the sampling distribution. This adaptability makes it more robust, especially in the high-dimensional setting characteristic of credit risk models.

- **Parallel computing capabilities:** MSMCS can leverage parallel computing, a significant advantage over traditional methods like Conditional Monte Carlo.

This allows for faster computations, making it more suitable for practical application where time efficiency is crucial.

- **Versatility in handling complex models:** MSMCS's ability to sample efficiently from complex, non-linear relationships ensures it can capture the intricate dynamics of credit markets—like in a Student's t copula.

8.6.1 MSMCS for Student's t copula

Following the Student's t copula framework established in Section 5.3.2, our goal is to determine the loss distribution from defaults in a portfolio within a specified time horizon. Here are the steps to apply MSMCS for this task.

Step 1: Model specification

1. **Portfolio of obligors:** Consider a portfolio of loans consisting of n obligors, each with a default probability $p_i \in (0, 1)$ and predetermined loss c_i in case of default.

2. **Latent variables:** Define a vector of latent variables $\mathbf{X} = (X_1, ..., X_n)$, where obligor i defaults if $X_i > x_i$, chosen so $\mathbb{P}(X_i > x_i) = p_i$.

3. **Portfolio loss function:** The portfolio loss from defaults is: $L(\mathbf{X}) = \sum_{i=1}^{n} c_i I_{X_i > x_i}$, where $I_{X_i > x_i} = 1$ if $X_i > x_i$, and 0 otherwise.

Step 2: Dependency structure with Student's t copula

1. **Common and idiosyncratic risk factors:** Assume the common risk factor, $\mathbf{Z}$, and individual idiosyncratic risks, η_i, are independent and Normally distributed. Specifically: $\mathbf{Z} \sim N(0, 1)$ and $\eta_i \sim N(0, \sigma_\eta^2)$ for $i = 1, ..., n$.

2. **Latent variable expression:** With a chosen probability $0 < p < 1$, the latent variables are expressed as: $X_i = \frac{p\mathbf{Z} + \sqrt{1-p^2}\eta_i}{T}$ for $i = 1, ..., n$, where T is

a non-negative random variable independent of $\mathbf{Z}$ and η_i.

3. **Student's t distribution:** For a positive integer k, let $T = \sqrt{k^{-1}\Gamma(1/2, k/2)}$, where Γ denotes the PDF of the Gamma distribution. Therefore, $\mathbf{X}$ follows a multivariate t-distribution with k degrees of freedom.

Step 3: Implementing MSMCS

1. **Initialistion:** Initialise MSMCS with the prior distributions for latent variables $\mathbf{X}$.

2. **Iterative process:** Through applying Algorithm 15, iteratively construct an equally weighted flat-histogram IS distribution, from which samples of $\mathbf{X}$ can be generated and the losses $L(\mathbf{X})$ calculated.

3. **Evaluate losses:** Consider the computed losses—from across the *bins*—to construct the overall loss distribution $L(\mathbf{X})$ for the portfolio; analyse the tail distribution and compute risk metrics, like VaR and CVaR.

8.6.2 Example application

We replicate the setup of Chan and Kroese (2010), setting $\sigma_\eta^2 = 9$, $x_i = \sqrt{n} \times 0.5$, $p_i = 0.25$, and $c_i = 1$ for all i. Both MMC-MCMC and MSMCS employ 20 iterations with 10^4 samples per iteration. As per numerical example four, we implement MMC-MCMC in two variations: one with a single long chain (MMC-MCMC-SC) and another with parallel chains (100 chains, each of length 100), the latter providing a parallel comparison to MSMCS. Additionally, we perform a standard MC simulation with varying sample sizes.

Results

Our primary interest lies in the probability of substantial losses, defined as $L(\mathbf{X}) > l$, with $l = bn$. We vary the degrees of freedom k or the sample size n, estimating the

probability of losses exceeding l for $b = 0.1, 0.2, 0.25, 0.3$. The results for varying the degrees of freedom k are given in Table 8.3 and for varying the sample size n in Table 8.4.

Table 8.3: Copula results using MC; MSMCS; and MMC-MCMC. (Part 1)

(a) $k = 4$ & $n = 250$

Large loss	**Sample size**	**Probability estimate**			
Threshold (b)	**MC**	**MC**	**MMC-MCMC-SC**	**MMC-MCMC-MC**	**MSMCS**
0.1	5×10^5	7.36×10^{-2}	7.27×10^{-2}	1.69×10^{-1}	7.31×10^{-2}
0.2	5×10^5	1.72×10^{-2}	1.63×10^{-2}	5.96×10^{-2}	1.71×10^{-2}
0.25	5×10^5	8.08×10^{-3}	8.13×10^{-3}	3.29×10^{-2}	8.05×10^{-3}
0.3	5×10^5	3.21×10^{-3}	3.24×10^{-3}	1.71×10^{-2}	3.28×10^{-3}

(b) $k = 8$ & $n = 250$

Large loss	**Sample size**	**Probability estimate**			
Threshold (b)	**MC**	**MC**	**MMC-MCMC-SC**	**MMC-MCMC-MC**	**MSMCS**
0.1	5×10^6	1.45×10^{-2}	1.39×10^{-2}	2.24×10^{-3}	1.42×10^{-2}
0.2	5×10^6	9.49×10^{-4}	9.43×10^{-4}	1.66×10^{-4}	9.49×10^{-4}
0.25	5×10^6	2.38×10^{-4}	2.49×10^{-4}	4.29×10^{-5}	2.46×10^{-4}
0.3	5×10^6	4.04×10^{-5}	3.98×10^{-5}	1.04×10^{-5}	4.01×10^{-5}

(c) $k = 12$ & $n = 250$

Large loss	**Sample size**	**Probability estimate**			
Threshold (b)	**MC**	**MC**	**MMC-MCMC-SC**	**MMC-MCMC-MC**	**MSMCS**
0.1	5×10^7	9.77×10^{-3}	9.82×10^{-3}	5.96×10^{-5}	9.78×10^{-3}
0.2	5×10^7	7.49×10^{-3}	7.63×10^{-3}	1.04×10^{-6}	7.53×10^{-3}
0.25	5×10^7	1.05×10^{-5}	1.02×10^{-5}	1.22×10^{-7}	1.03×10^{-5}
0.3	5×10^7	1.12×10^{-6}	1.34×10^{-6}	1.65×10^{-8}	1.21×10^{-6}

(d) $k = 16$ & $n = 250$

Large loss	**Sample size**	**Probability estimate**			
Threshold (b)	**MC**	**MC**	**MMC-MCMC-SC**	**MMC-MCMC-MC**	**MSMCS**
0.1	5×10^8	9.40×10^{-4}	9.36×10^{-4}	2.50×10^{-6}	9.43×10^{-4}
0.2	5×10^8	6.91×10^{-6}	6.90×10^{-6}	9.58×10^{-9}	6.86×10^{-6}
0.25	5×10^8	6.22×10^{-7}	6.18×10^{-7}	6.04×10^{-10}	6.19×10^{-7}
0.3	5×10^8	4.40×10^{-8}	4.37×10^{-8}	3.67×10^{-11}	4.51×10^{-8}

(e) $k = 20$ & $n = 250$

Large loss	**Sample size**	**Probability estimate**			
Threshold (b)	**MC**	**MC**	**MMC-MCMC-SC**	**MMC-MCMC-MC**	**MSMCS**
0.1	5×10^8	2.83×10^{-4}	2.88×10^{-4}	1.39×10^{-7}	2.76×10^{-4}
0.2	5×10^8	7.98×10^{-7}	7.61×10^{-7}	1.35×10^{-10}	7.73×10^{-7}
0.25	5×10^8	5.40×10^{-8}	4.92×10^{-8}	2.99×10^{-12}	5.32×10^{-8}
0.3	5×10^8	0	5.72×10^{-9}	1.02×10^{-13}	5.63×10^{-9}

Table 8.4: Copula results using MC; MSMCS; and MMC-MCMC. (Part 2)

(a) $k = 12$ & $n = 250$

Large loss Threshold (b)	Sample size	Probability estimate			
	MC	MC	MMC-MCMC-SC	MMC-MCMC-MC	MSMCS
0.1	5×10^7	9.77×10^{-3}	9.82×10^{-3}	5.96×10^{-5}	9.78×10^{-3}
0.2	5×10^7	7.49×10^{-3}	7.63×10^{-3}	1.04×10^{-6}	7.53×10^{-3}
0.25	5×10^7	1.05×10^{-5}	1.02×10^{-5}	1.22×10^{-7}	1.03×10^{-5}
0.3	5×10^7	1.12×10^{-6}	1.34×10^{-6}	1.65×10^{-8}	1.21×10^{-6}

(b) $k = 12$ & $n = 500$

Large loss Threshold (b)	Sample size	Probability estimate			
	MC	MC	MMC-MCMC-SC	MMC-MCMC-MC	MSMCS
0.1	5×10^8	9.61×10^{-5}	9.42×10^{-5}	5.08×10^{-12}	9.52×10^{-5}
0.2	5×10^8	1.34×10^{-6}	1.39×10^{-6}	7.15×10^{-13}	1.38×10^{-6}
0.25	5×10^8	1.36×10^{-7}	1.57×10^{-7}	4.37×10^{-13}	0.84×10^{-7}
0.3	5×10^8	1.00×10^{-8}	1.29×10^{-8}	2.54×10^{-13}	1.27×10^{-8}

(c) $k = 12$ & $n = 1000$

Large loss Threshold (b)	Sample size	Probability estimate			
	MC	MC	MMC-MCMC-SC	MMC-MCMC-MC	MSMCS
0.1	3×10^8	1.96×10^{-6}	1.88×10^{-6}	2.54×10^{-13}	1.91×10^{-6}
0.2	3×10^8	3.67×10^{-8}	3.58×10^{-8}	6.29×10^{-14}	3.72×10^{-8}
0.25	3×10^8	2.39×10^{-9}	2.24×10^{-9}	4.18×10^{-14}	2.28×10^{-9}
0.3	3×10^8	0	3.25×10^{-10}	7.24×10^{-15}	3.19×10^{-10}

Interpretatation of results

As MMC reconstructs the whole loss distribution, we only require one simulation to be performed from which the loss probability for any b-value can be obtained. This is a significant computational saving compared to other existing methods, like the conditional-MC proposed by Chan and Kroese (2010), which require a new simulation for each b-value.

Compared to standard MC, MMC is very effective for the copula model estimation of large loss probabilities. Both MMC-MCMC (with a single long chain, denoted by MMC-MCMC-SC) and MSMCS are very effective here—although MMC-MCMC (with multiple parallel chains, denoted by MMC-MCMC-MC) performs poorly, particularly in the high-dimensional setting, clearly illustrating the advantage of MSMCS in the parallel implementation.

8.6.3 Conclusion

In conclusion, MSMCS offers a new approach to credit risk modelling, utilising a Student's t copula to offer: efficient sampling of tail probabilities, adaptive importance sampling, parallel computing capabilities, and versatility in handling complete models. MSMCS provide a more accurate estimation of tail risks, contributing to more robust risk assessment strategies and aiding financial institutions in better preparing for extreme market events. With a more accurate credit risk model, financial institutions can allocate capital more efficiently, ensuring they are adequately prepared for potential losses.

8.7 Assessing equity risk using MSMCS

In the context of equity risk (detailed in Section 5.2), understanding and managing the volatility of stock prices and the potential for significant market movements is critical, especially when considering the impact of 'tail events'. VaR provides a threshold value such that the probability of a loss exceeding this threshold is at a given level. However, VaR has limitations, particularly its failure to capture the magnitude of potential losses beyond its threshold. CVaR addresses this by representing the expected loss, given that the losses are worse than the VaR value. CVaR offers a more comprehensive view of potential losses in the tail of the distribution.

Importance sampling can be used for CVaR approximation, and whilst IS is more efficient than Monte Carlo for CVaR estimation, it still has limitations. Xing et al. (2022) propose applying IS within a nested simulation framework to reduce the variance in the CVaR estimation. Deo and Murthy (2021) propose a black-box importance sampling for VaR and CVaR estimation. Finally, Heinkenschloss et al. (2018) propose a reduced-order model approach to CVaR estimation. For further details on importance sampling for CVaR estimation, refer to the works of Sun and

Hong (2009).

In Chapter 4, we introduced a Bayesian framework for ARMA-GARCH models to forecast future asset returns. This framework combines ARMA's ability to predict the mean of a series with GARCH's ability to model and forecast changing volatility. The Bayesian approach results in a posterior distribution for the ARMA-GARCH forecast. By reconstructing this distribution, we can obtain a distribution of forecasted returns—from which the expected return and risk measures (like VaR and CVaR) can be obtained.

This section shows how MSMCS can be applied to reconstruct the forecasted returns distribution from the Bayesian ARMA-GARCH model. The MSMCS approach shares the same four advantages as the credit risk application: efficient sampling of tail probabilities, adaptive importance sampling, parallel computing capabilities, and versatility in handling complex models.

8.7.1 MSMCS for ARMA-GARCH model

Problem set-up

1. **Portfolio weights:** Define the N-dimensional vector $\mathbf{w} = (w_1, ..., w_N)$ representing the allocation to each of the i^{th} asset. The weights must be non-negative, and their sum cannot exceed one.

2. **Forecast function:** Define $f(\mathbf{w}, \mathbf{z})$ to represent the forecasted returns for a portfolio with weights $\mathbf{w}$ and realisation $\mathbf{z}$ of $\mathbf{Z}$. $\mathbf{z}$ represents one realisation of the Bayesian ARMA-GARCH model.

Bayesian ARMA and GARCH models

1. **ARMA model:** Using the process in Chapter 4, determine the optimal ARMA model order choice and fit a Bayesian ARMA model to the histori-

cal time series for each asset i.

2. **GARCH model:** Using the process in Chapter 4, determine the optimal GARCH model order choice and fit a Bayesian GARCH model to the ARMA-residuals for each asset i by choosing the appropriate prior distribution for the GARCH parameters.

3. **Likelihood functions:** Construct likelihood functions for both models, reflecting the probability of observing return data given the model parameters.

4. **Predictive posterior distribution:** Combine the priors and likelihoods to define predictive posterior distributions for the ARMA-GARCH model, representing the forecasted asset returns, namely $\mathbf{Z}$.

Implementing MSMCS

1. **Initialisation:** Initialise MSMCS using the prior ARMA-GARCH distributions for a set portfolio $\mathbf{w}$.

2. **Iterative process:** Through applying Algorithm 15, iteratively construct an equally weighted flat-histogram IS distribution to generate samples from the Bayesian ARMA-GARCH model representing forecasts $f(\mathbf{w}, \mathbf{z})$.

3. **Evaluate losses:** Use the forecasted values from the overall forecasted return distribution $f(\mathbf{w}, \mathbf{Z})$; compute the overall return expectation and assess the tail distribution to compute risk measures like VaR and CVaR.

8.7.2 Example application

To demonstrate the performance of the Bayesian ARMA-GARCH framework combined with MSMCS, we implement it on our core data. The Bayesian ARMA-GARCH models are as defined in Section 4.5.1. To provide a comparison, we also

implement MSMCS on the frequentist ARMA-GARCH framework. The standard frequentist implementation of ARMA models obtains a single estimate of a forecasted mean return. The frequentist GARCH model obtains a single estimate of forecasted return variance. Assuming that the returns follow a Normal distribution—typical in frequentist ARMA-GARCH models—we can obtain the forecasted return distribution using these forecasted means and variances, with MSMCS.

Using the historical data, we identified the biggest 20-day bull, bear, and volatile periods for the S&P 500, FTSE 100, and SSE indexes—and applied our framework over these dates by country. The results follow.

Forecasting in a bull market for US assets

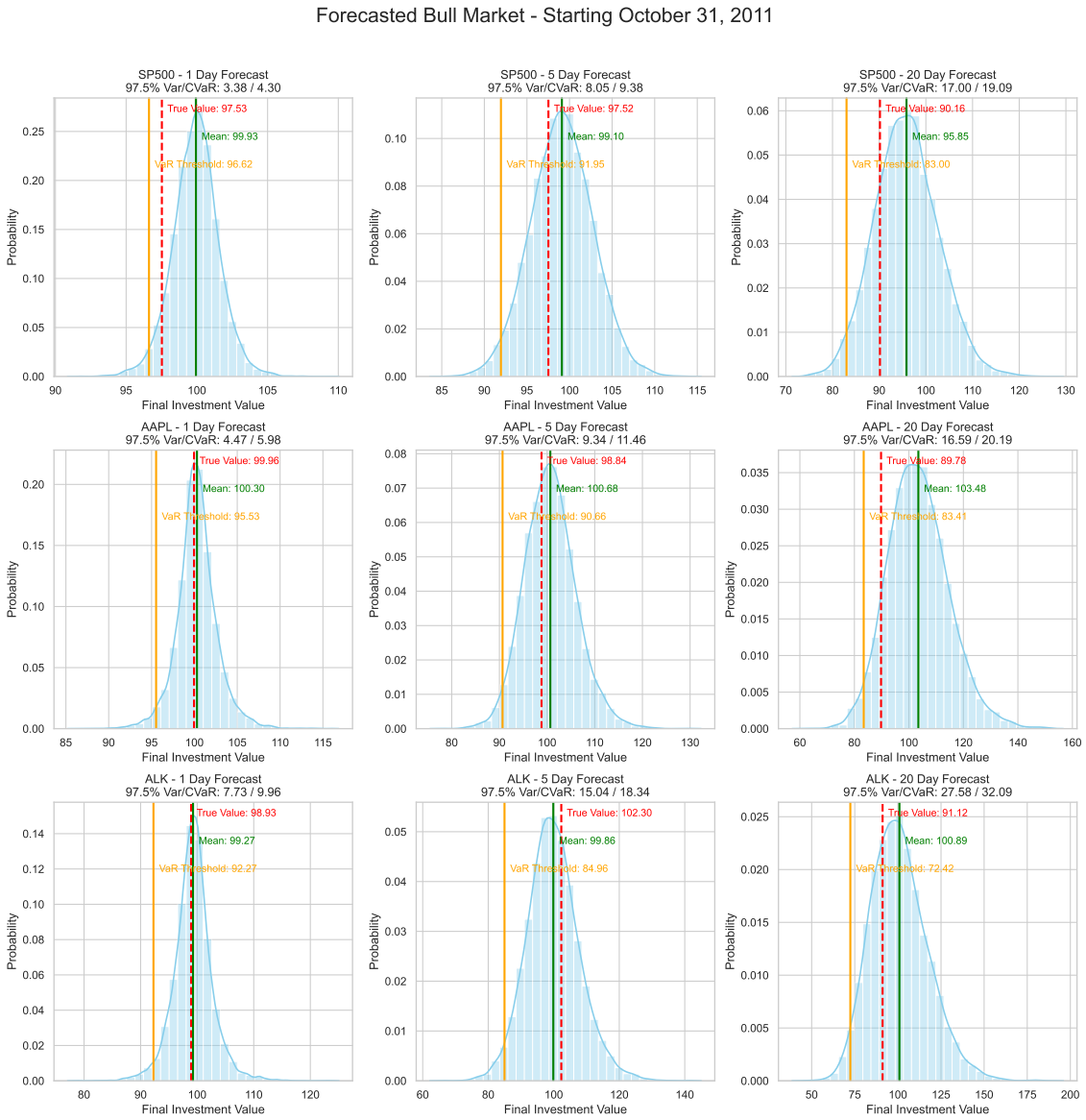


Figure 8.9: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bull market (US)

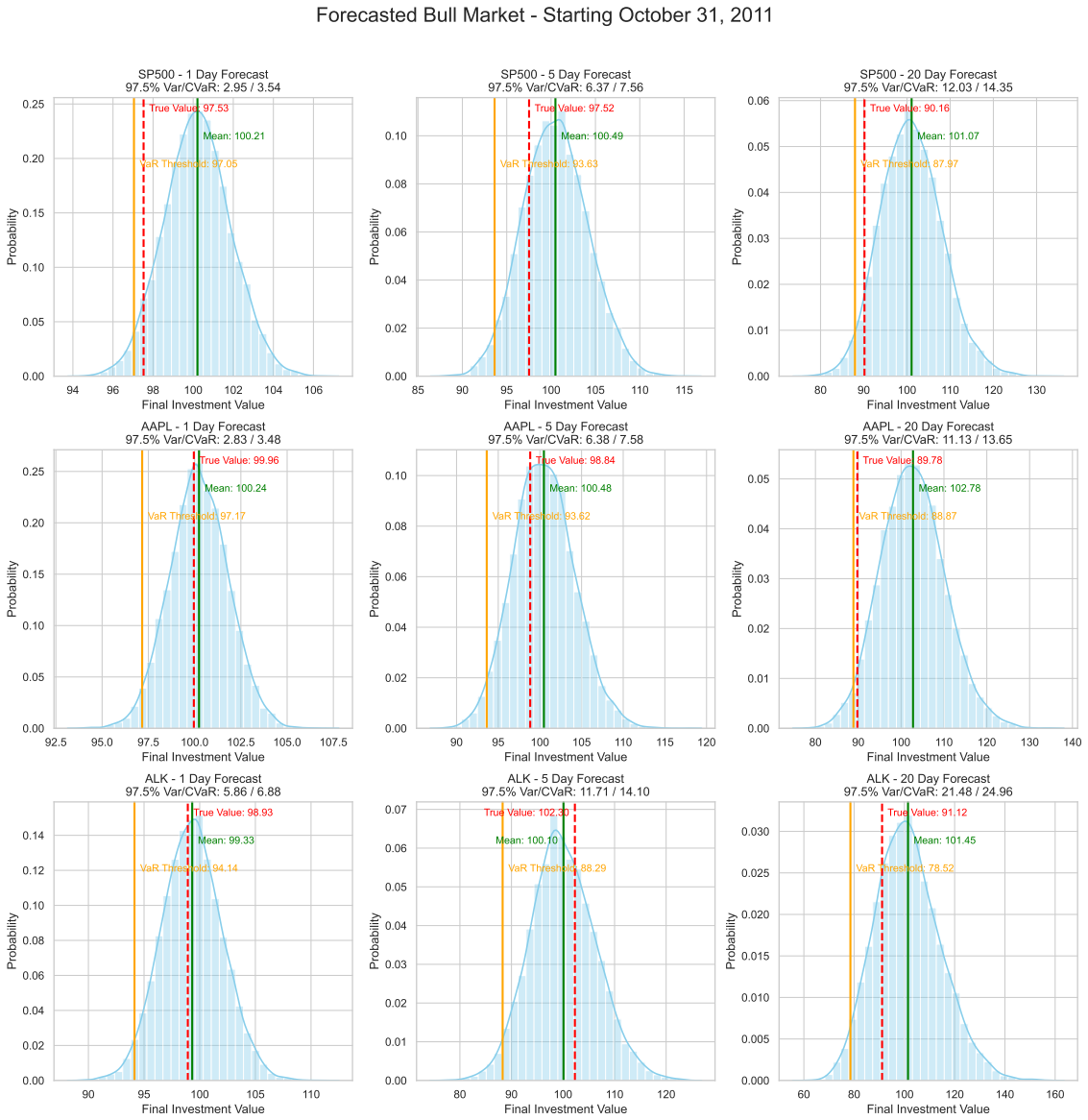


Figure 8.10: Return distribution using MSMCS on frequentist ARMA-GARCH Forecasts: Bull market (US)

Forecasting in a bull market for UK assets

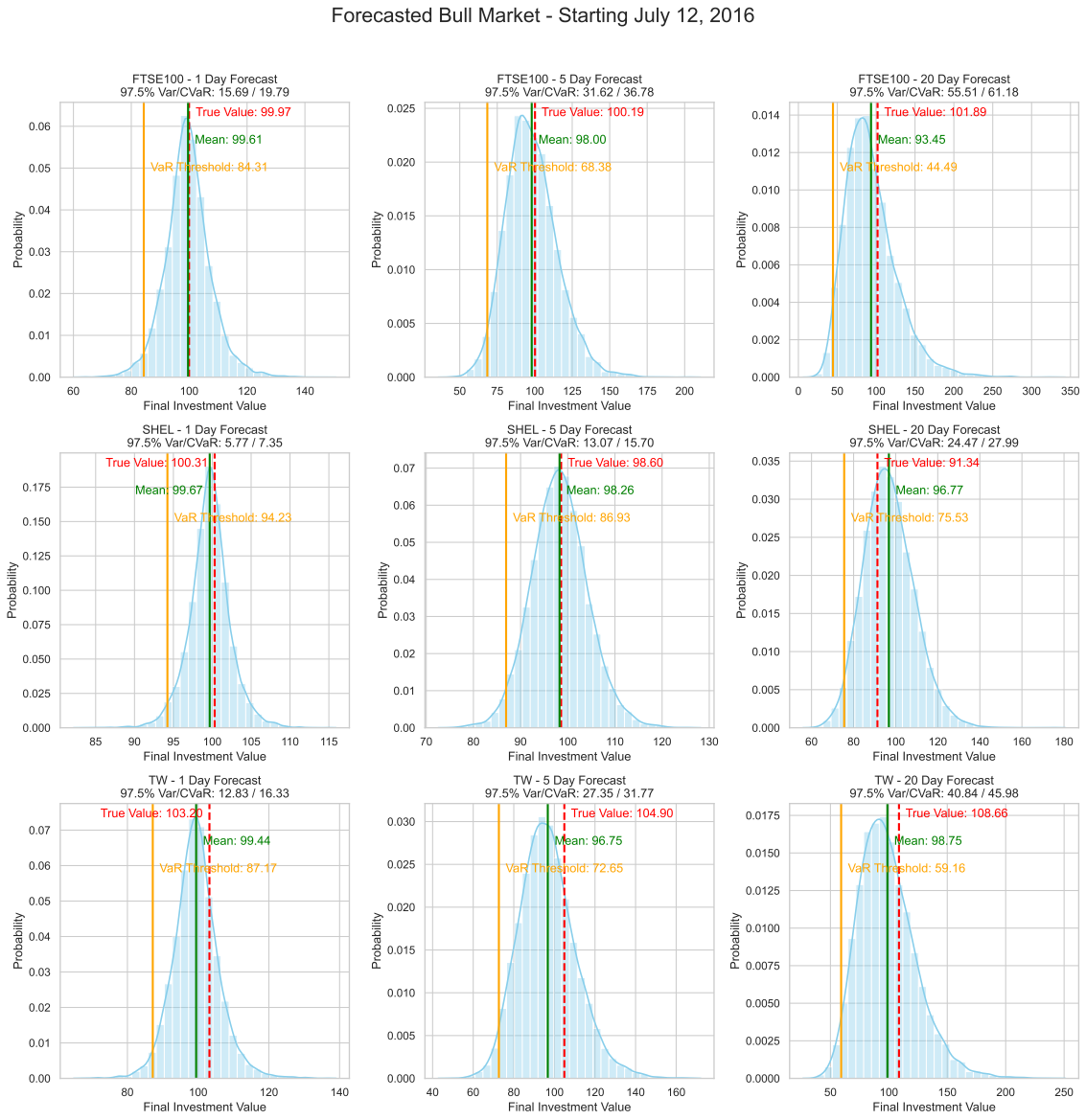


Figure 8.11: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bull market (UK)

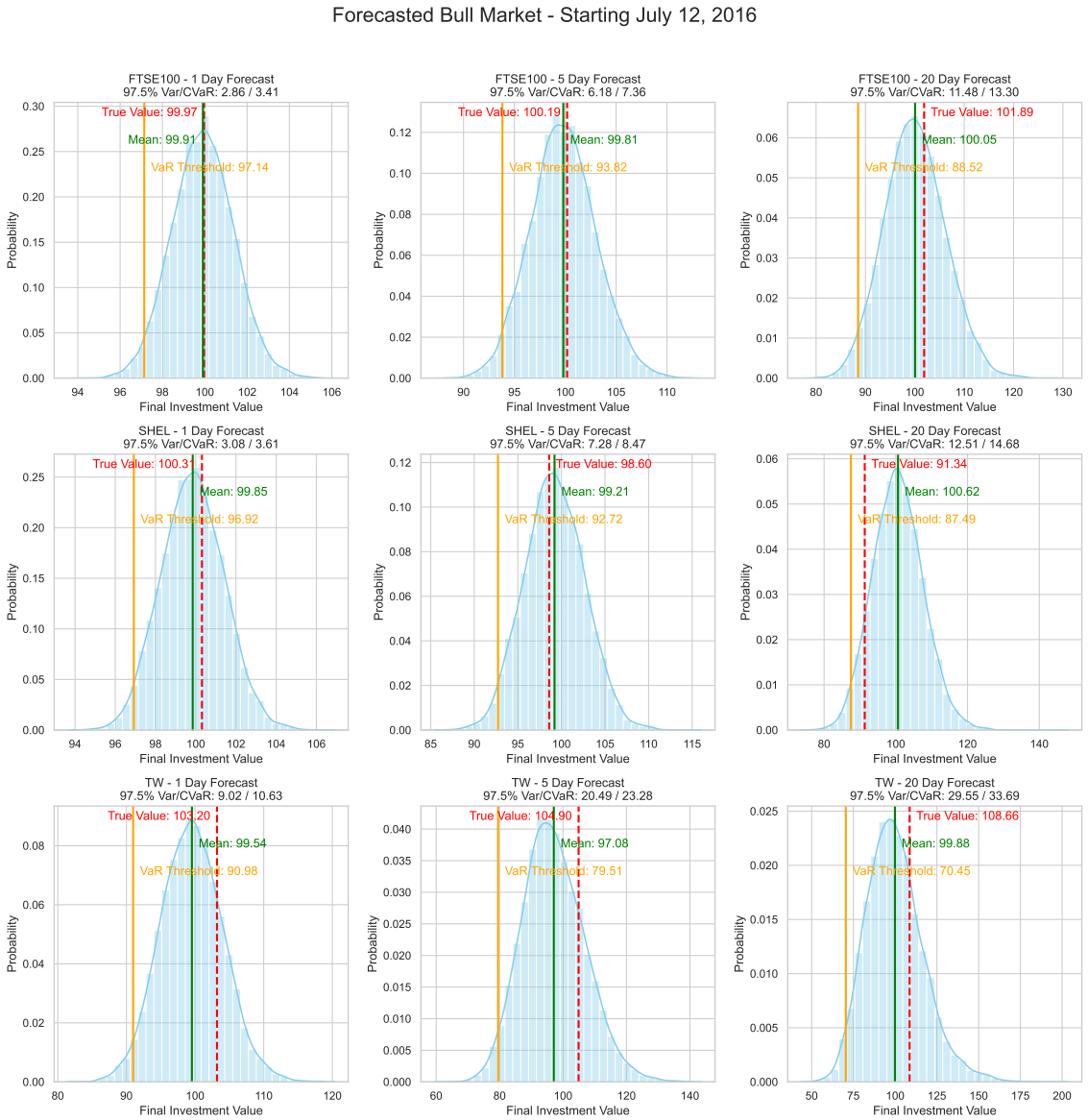


Figure 8.12: Return distribution using MSMCS on frequentist ARMA-GARCH Forecasts: Bull market (UK)

Forecasting in a bull market for CN assets

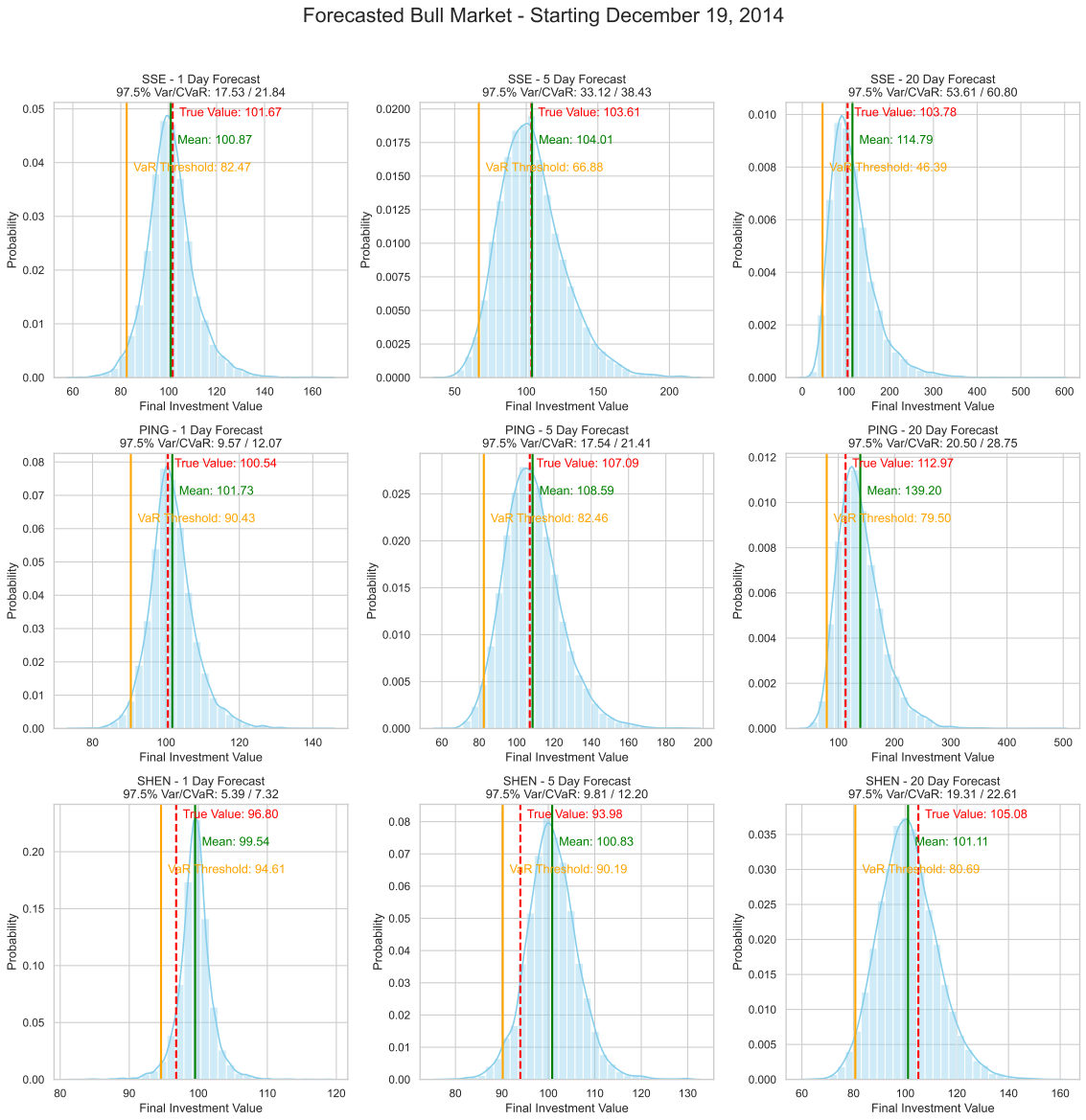


Figure 8.13: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bull market (CN)

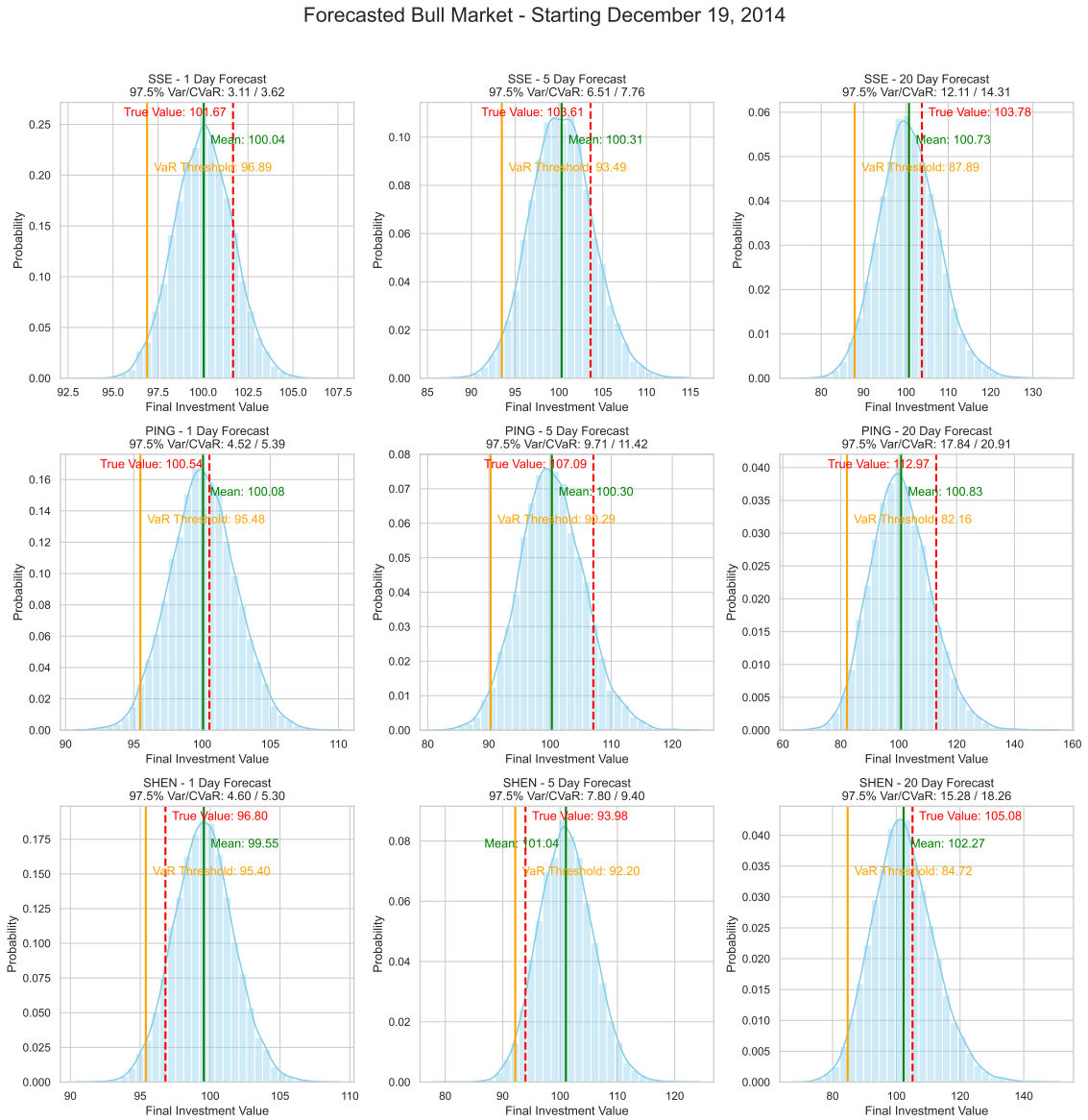


Figure 8.14: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Bull market (CN)

Forecasting in a bear market for US assets

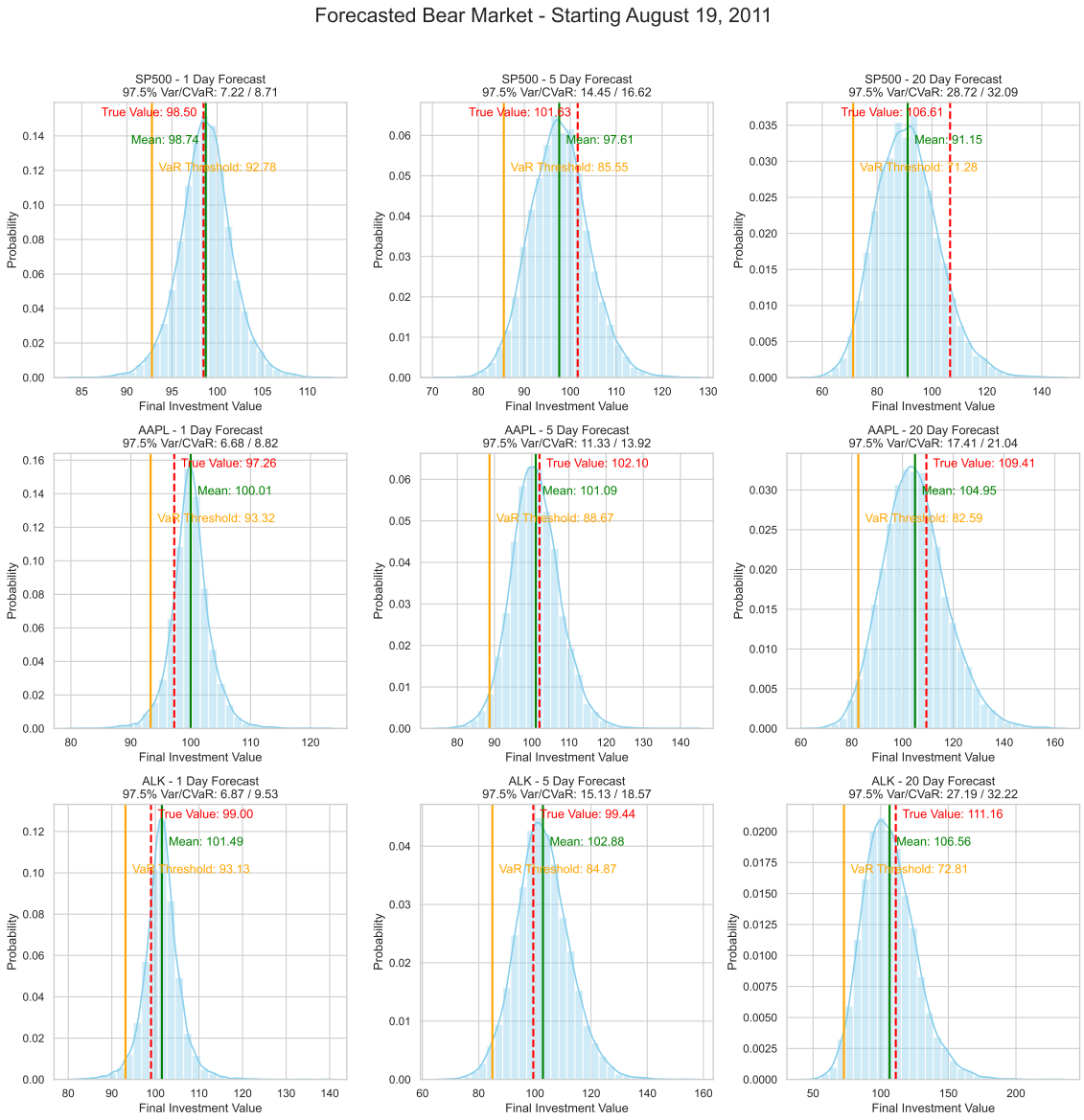


Figure 8.15: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bear market (US)

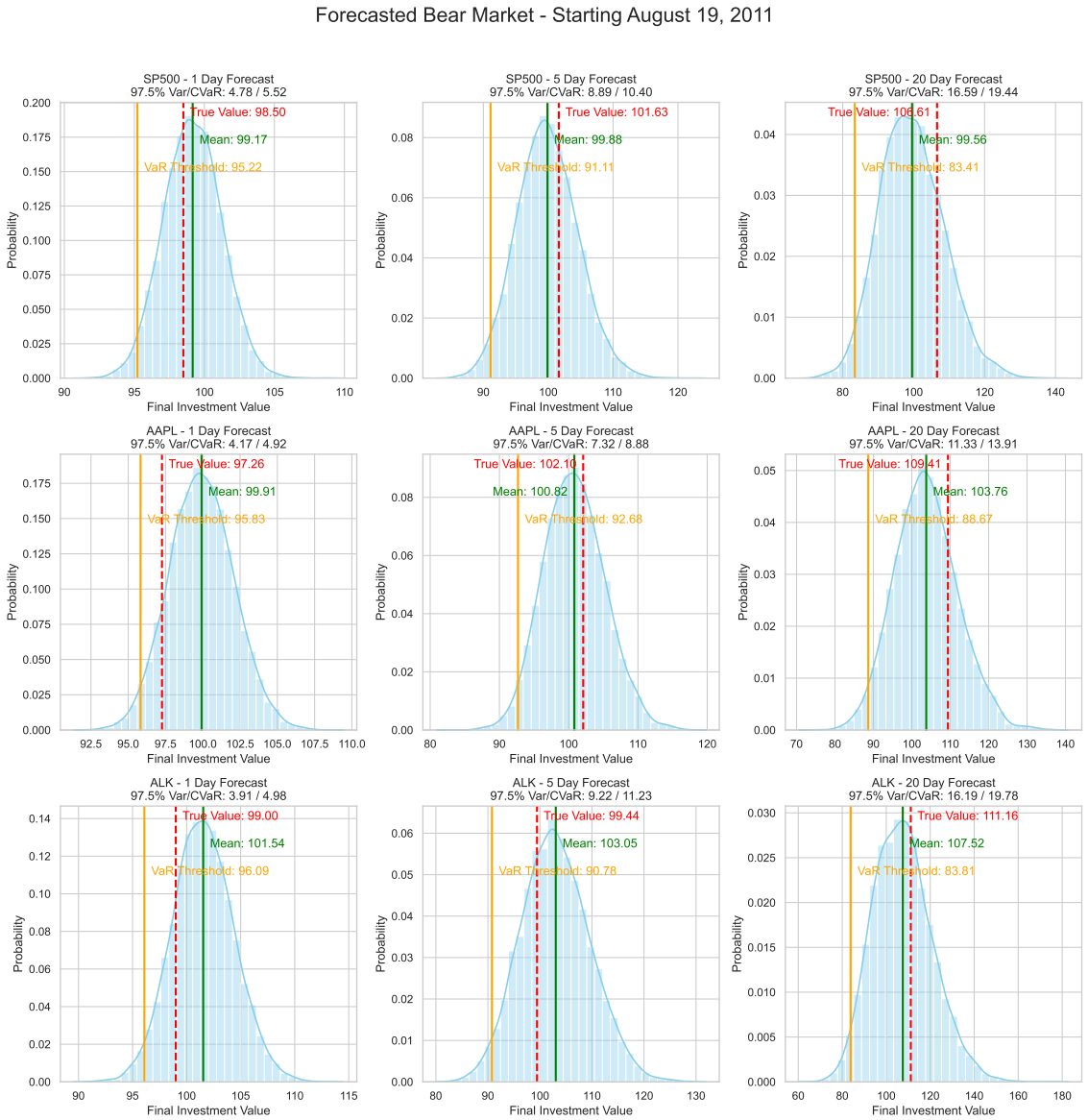


Figure 8.16: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Bear market (US)

Forecasting in a bear market for UK assets

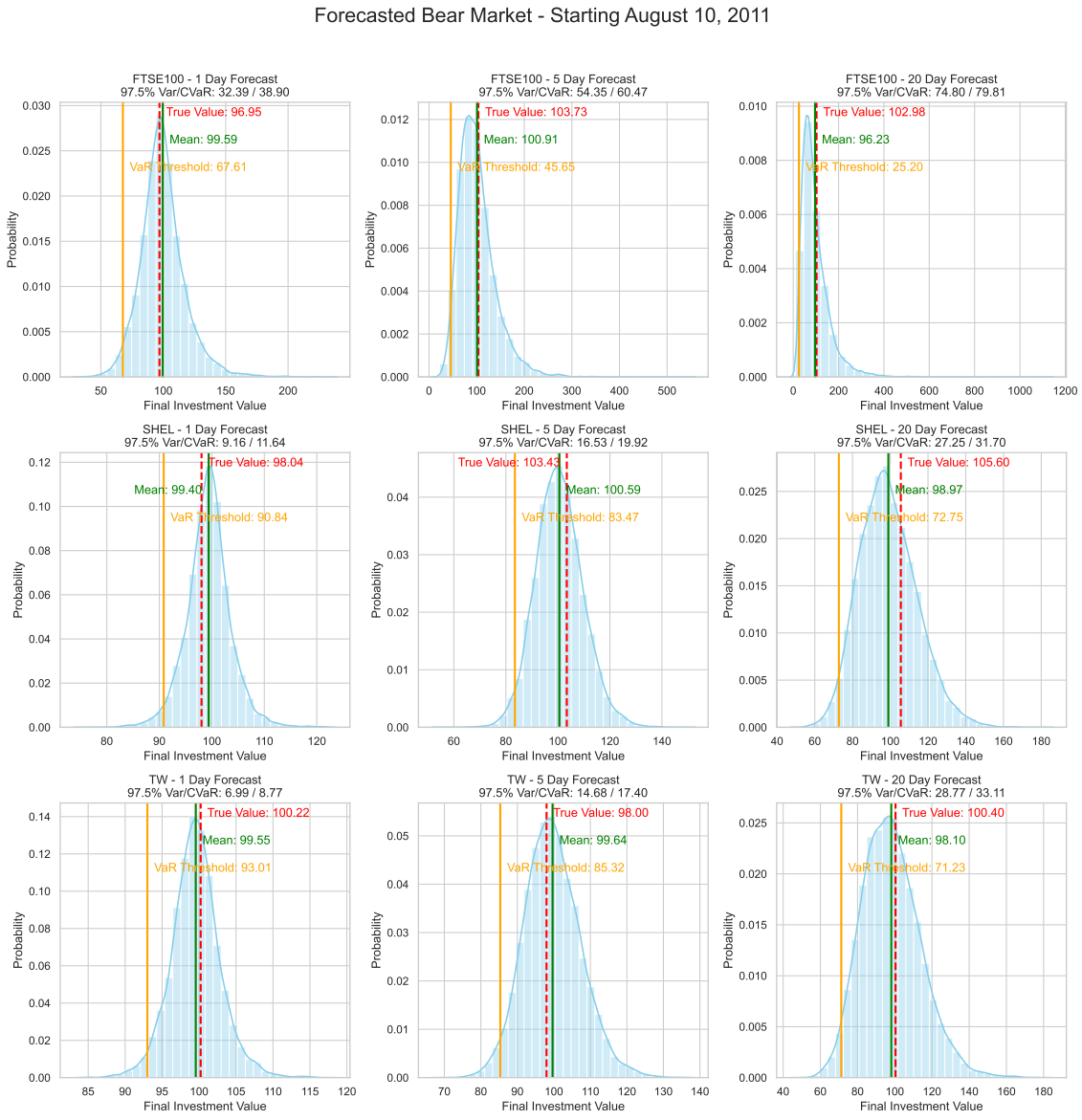


Figure 8.17: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bear market (UK)

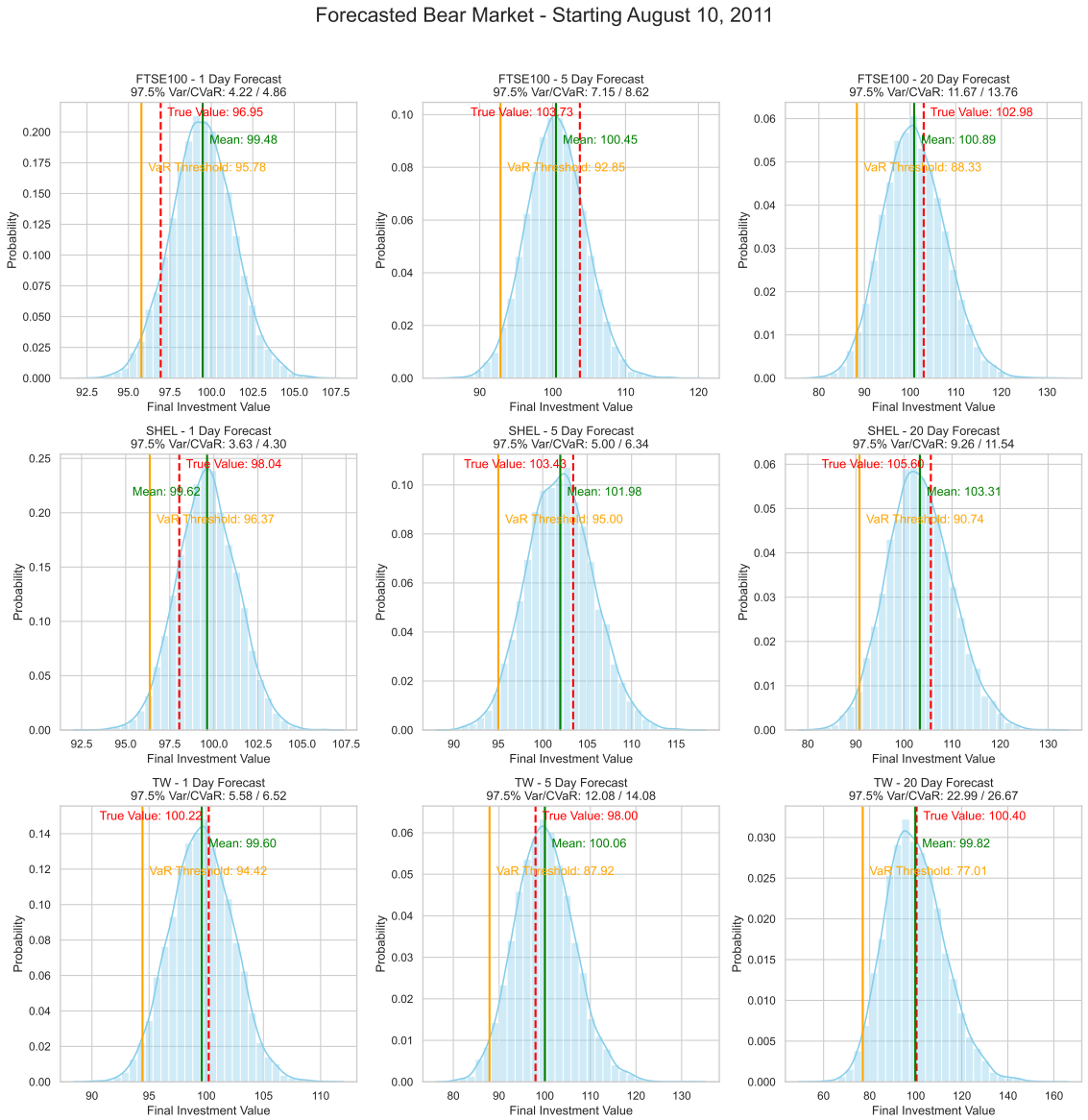


Figure 8.18: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Bear market (UK)

Forecasting in a bear market for CN assets

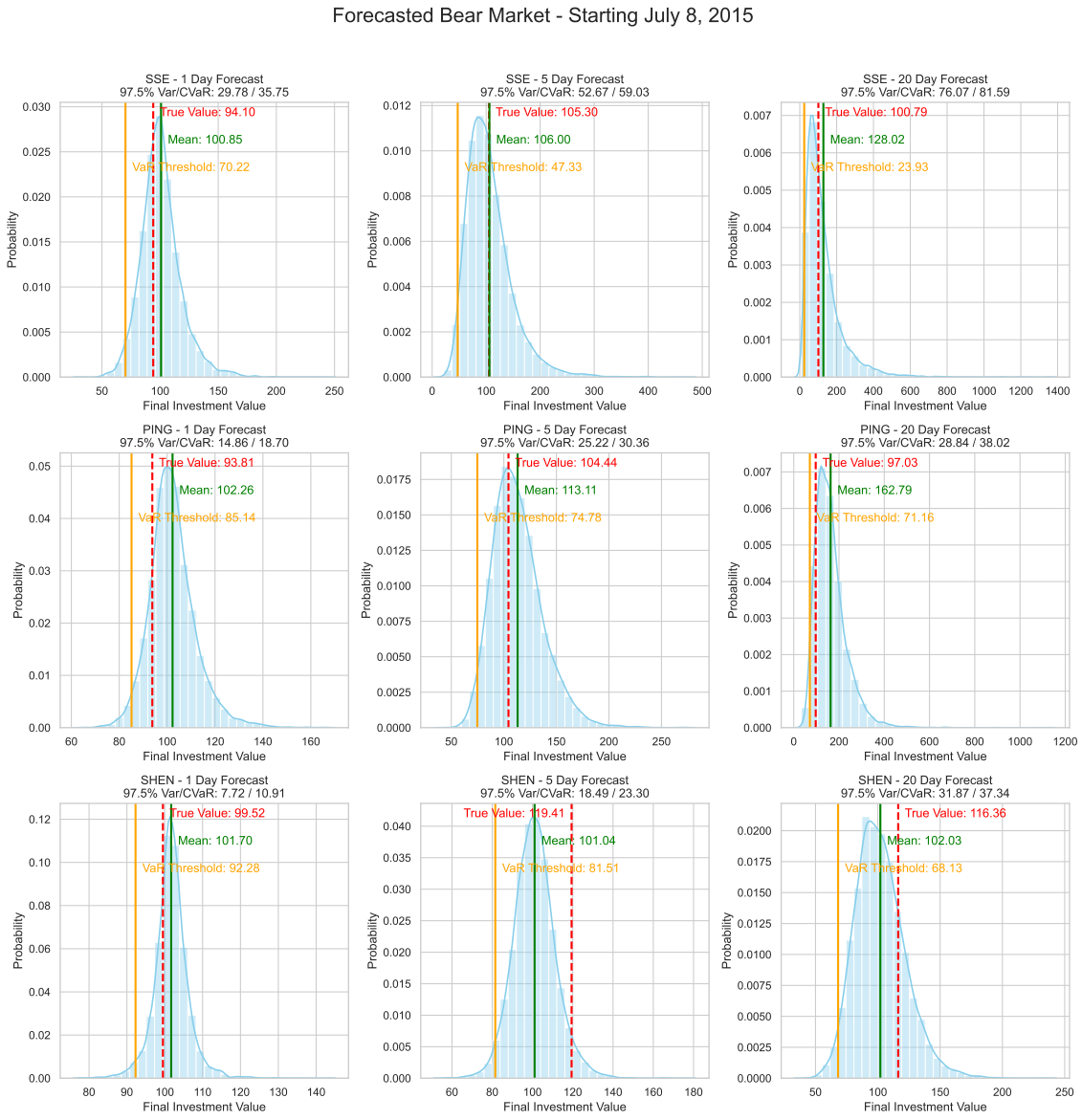


Figure 8.19: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Bear market (CN)

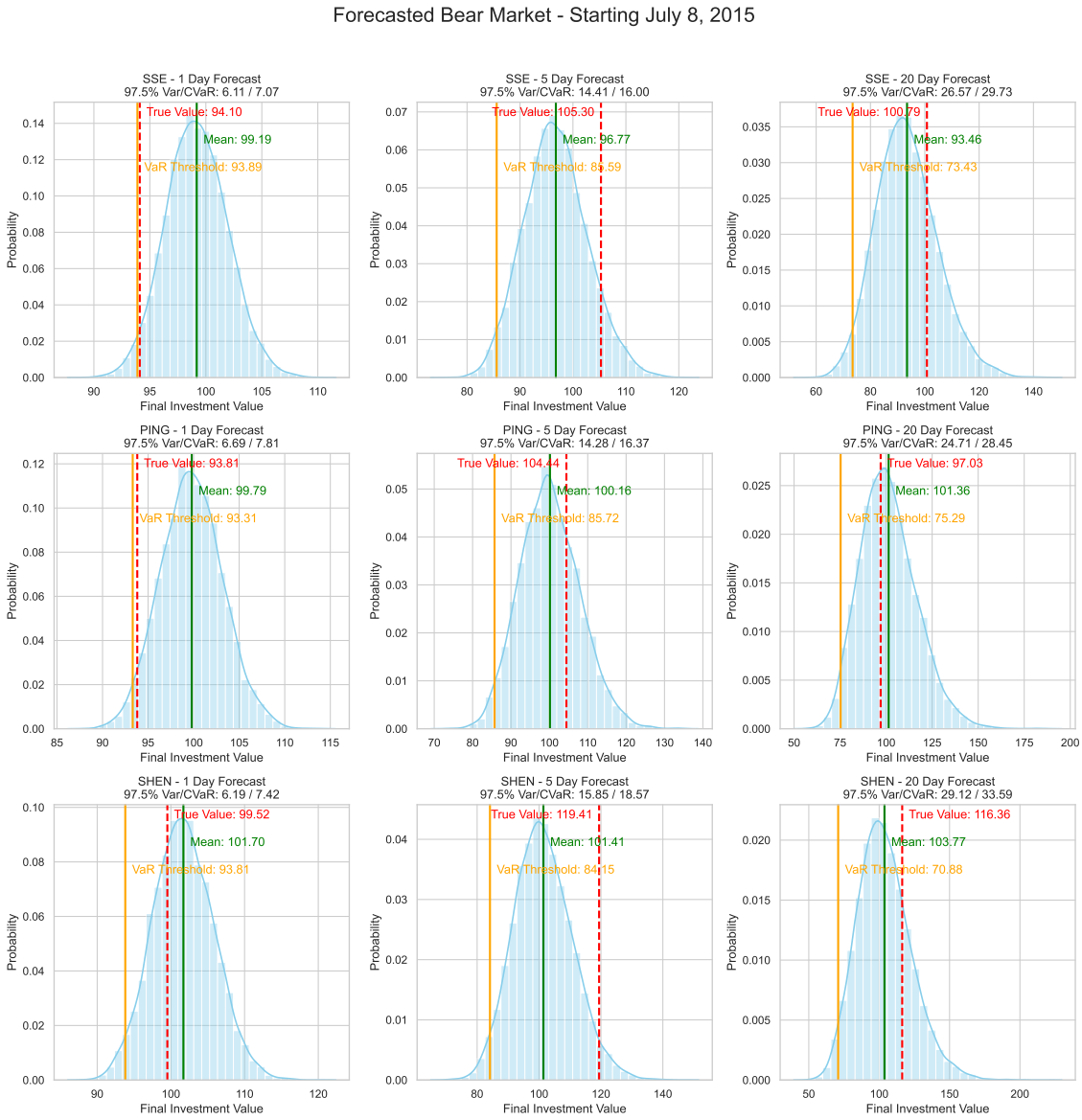


Figure 8.20: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Bear market (CN)

Forecasting in a volatile market for US assets

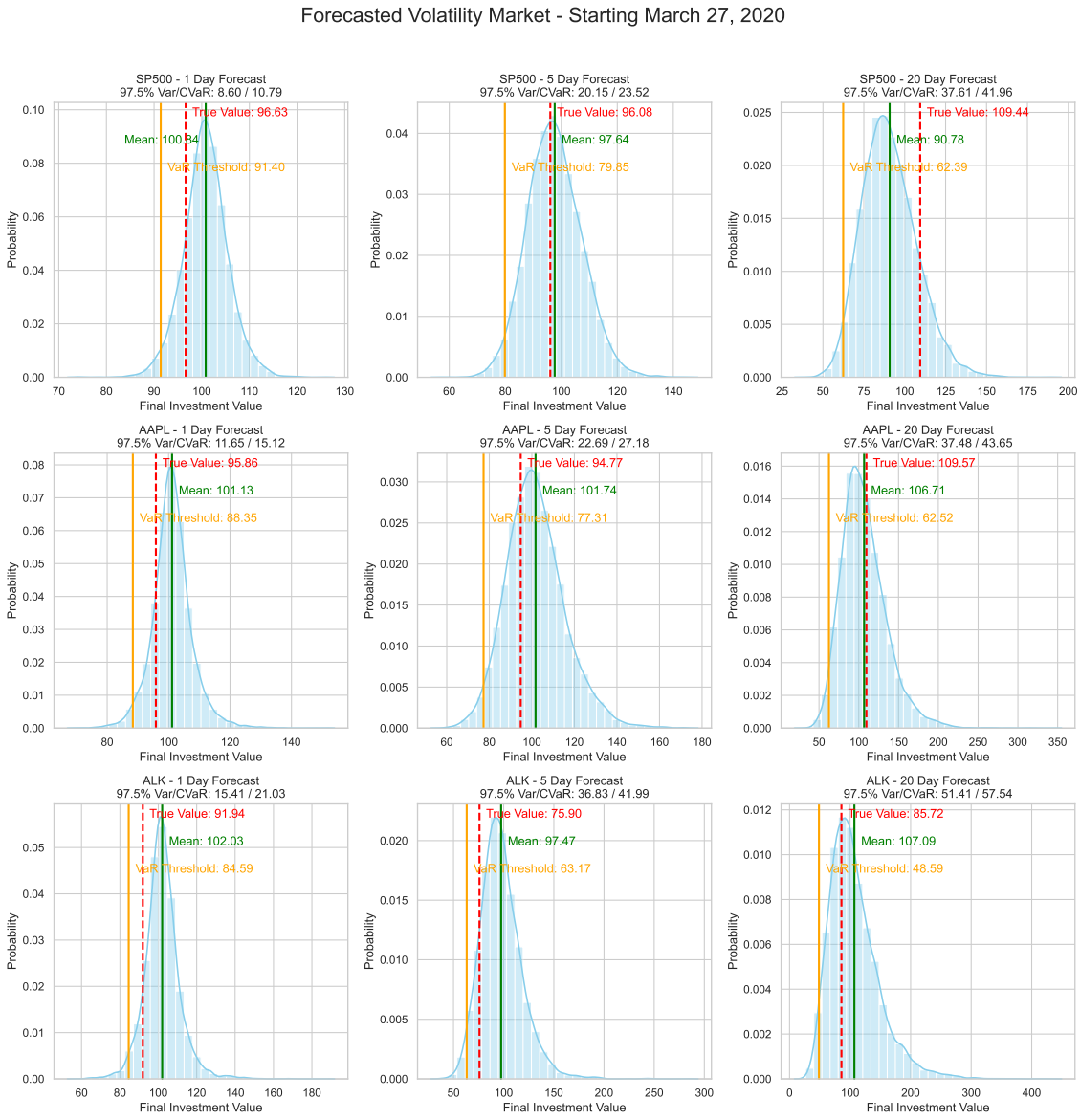


Figure 8.21: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Volatile market (US)

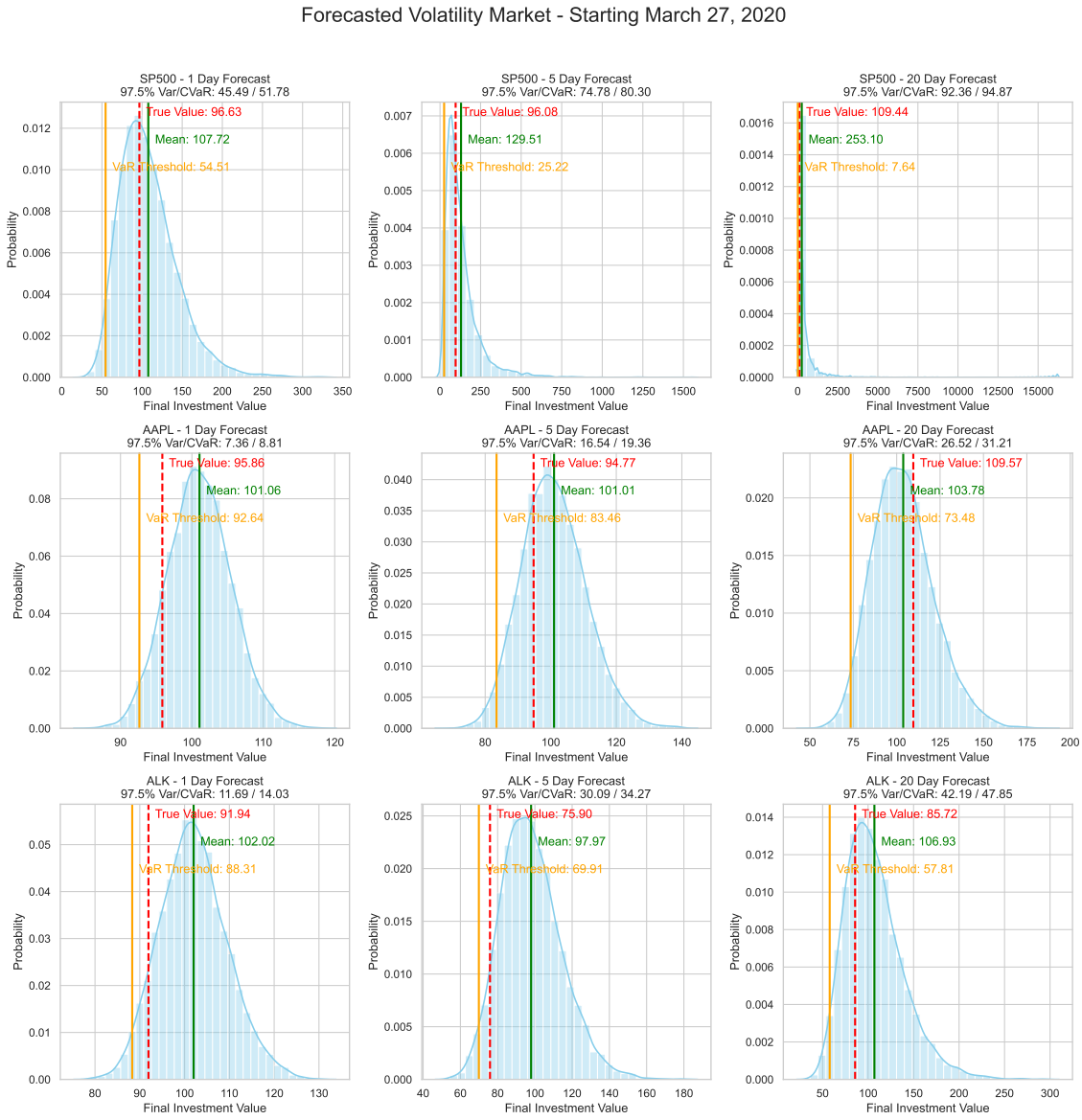


Figure 8.22: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Volatile market (US)

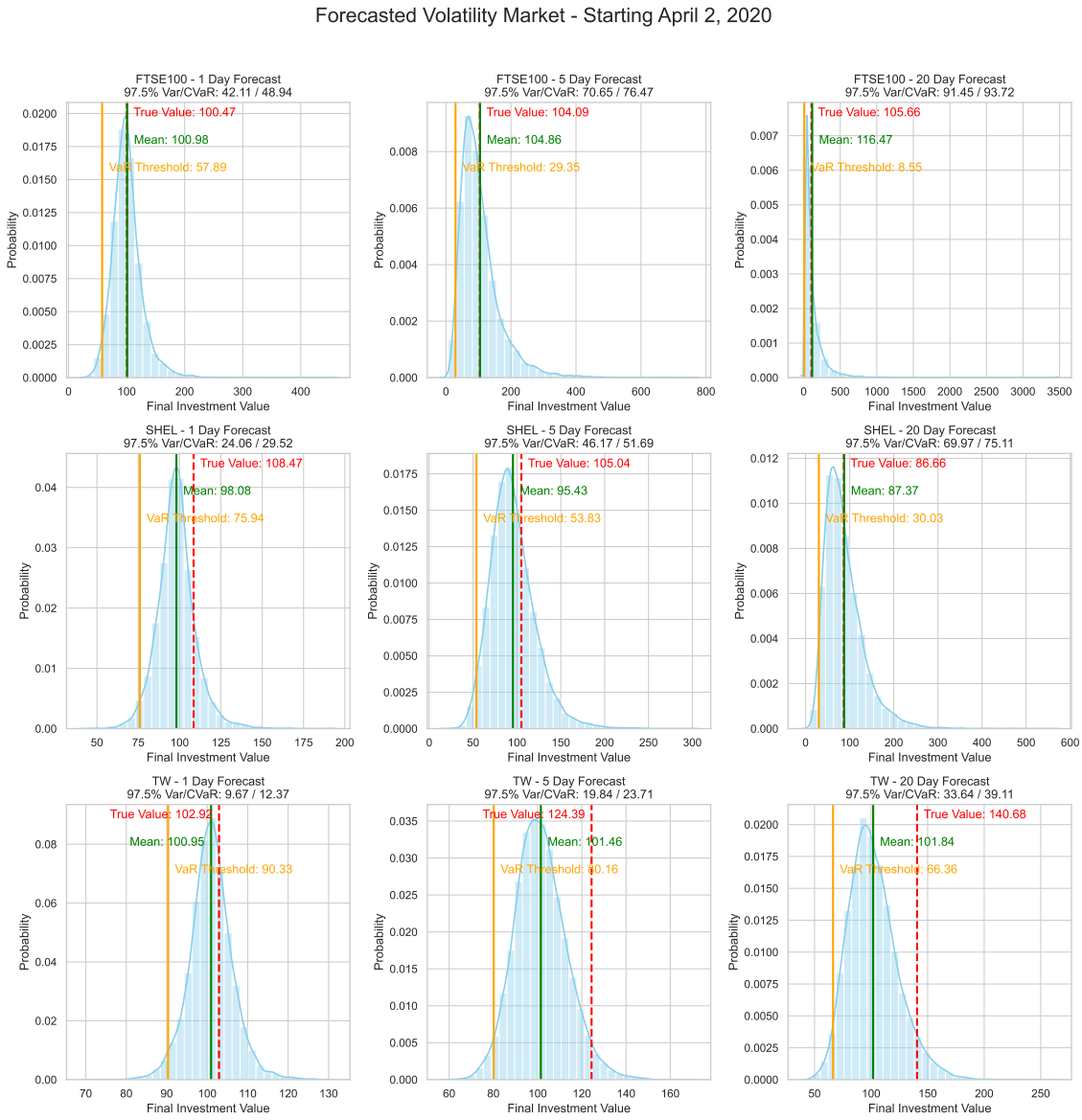


Figure 8.23: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Volatile market (UK)

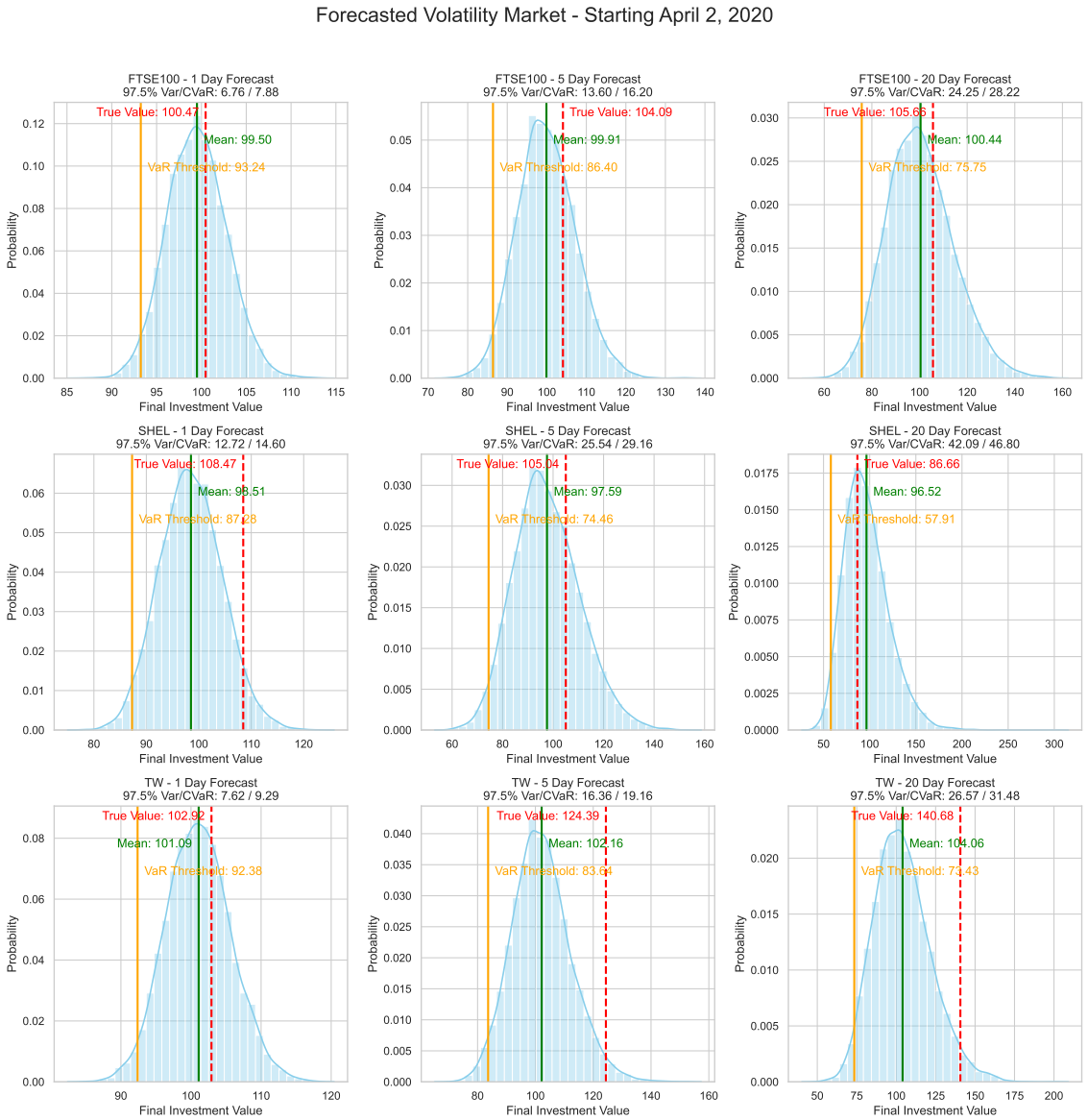


Figure 8.24: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Volatile market (UK)

Forecasting in a volatile market for CN assets

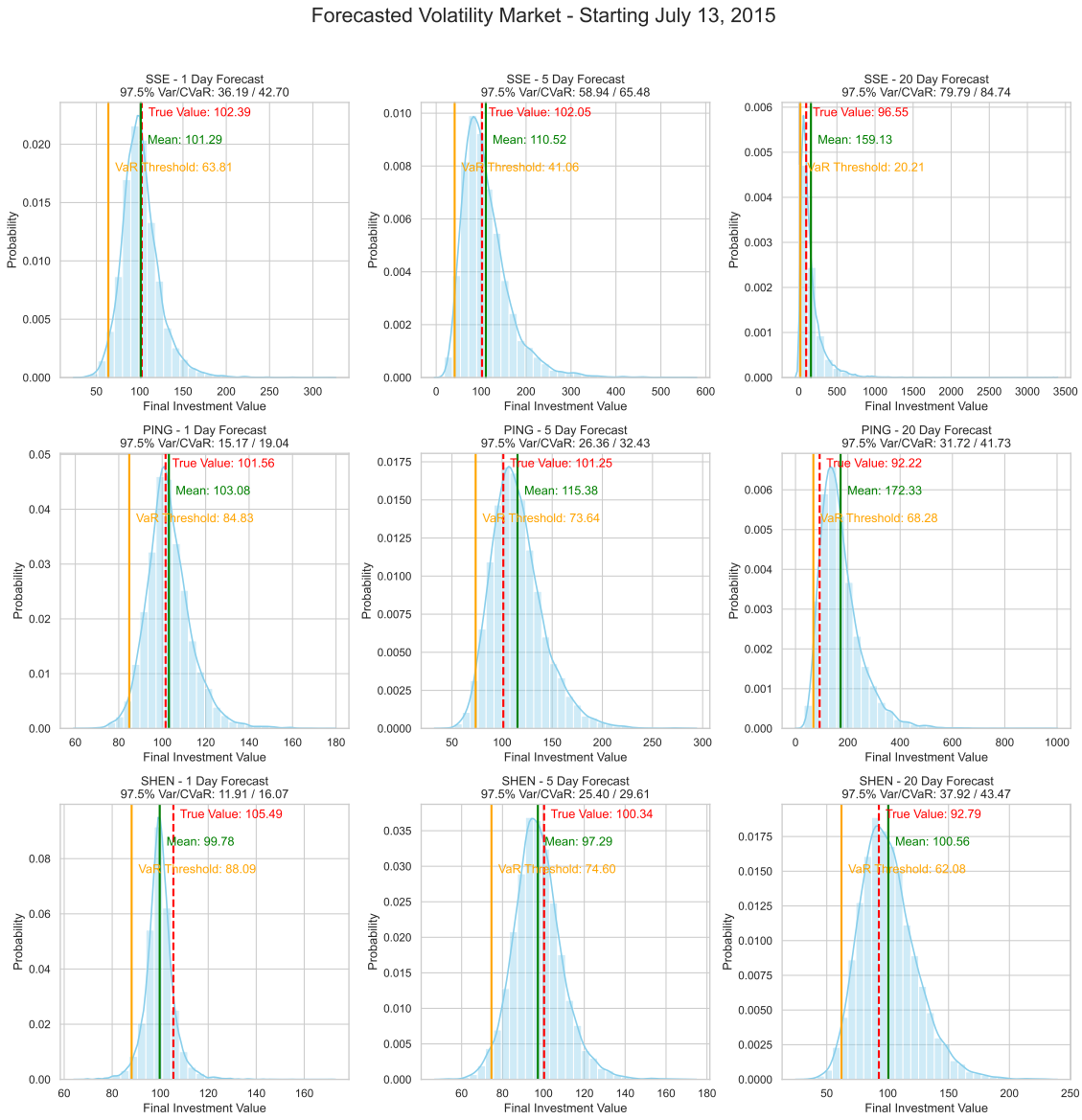


Figure 8.25: Return distribution using MSMCS on Bayesian ARMA-GARCH forecasts: Volatile market (CN)

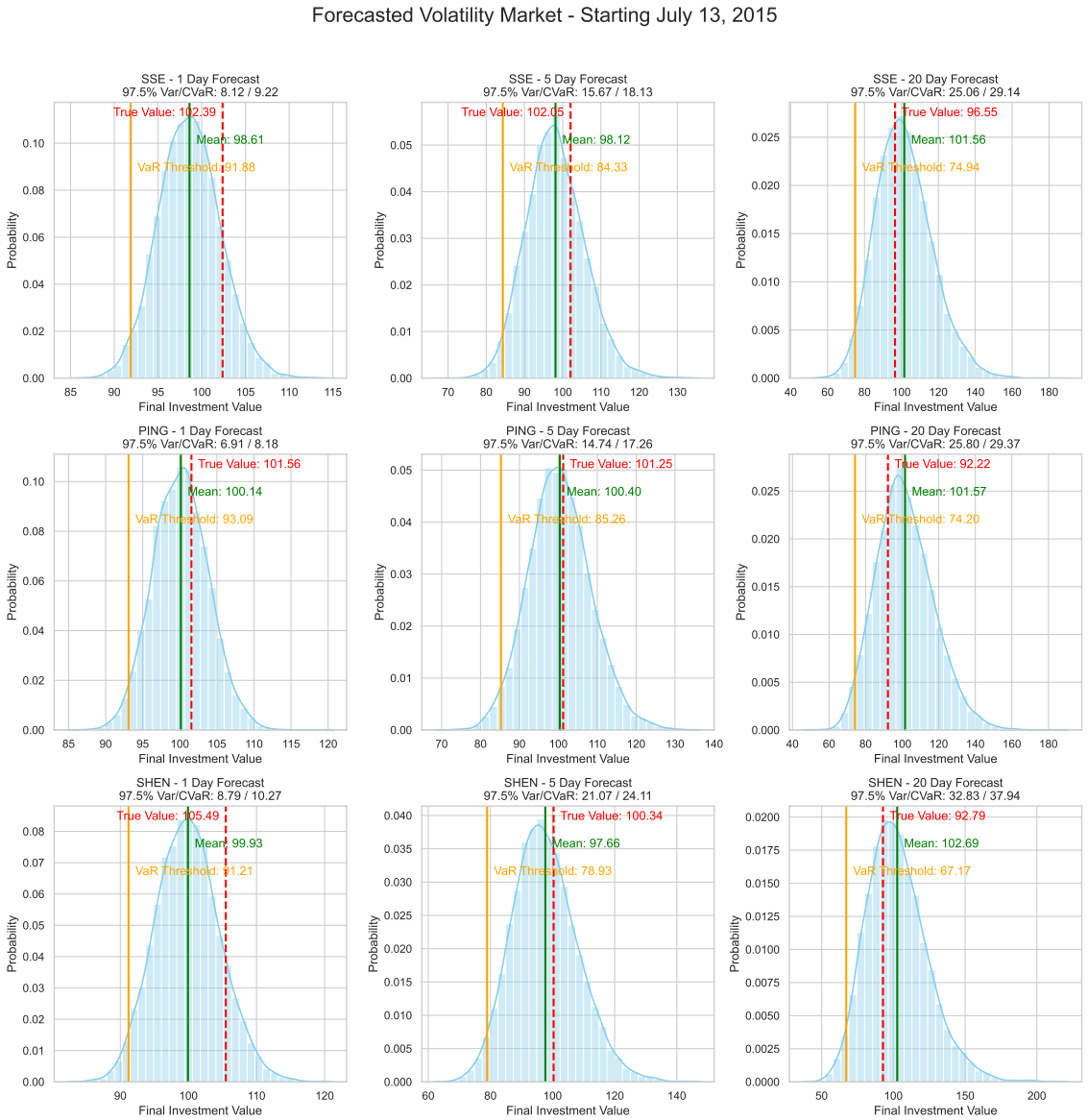


Figure 8.26: Return distribution using MSMCS on frequentist ARMA-GARCH forecasts: Volatile market (CN)

8.7.3 Conclusion

Applying MSMCS in equity risk assessment significantly enhances our understanding and capabilities in financial risk management. The MSMCS approach, applied to both Bayesian and frequentist ARMA-GARCH models, can efficiently reconstruct a complete probability distribution and obtain a CVaR estimate. Based on our core data, the Bayesian ARMA-GARCH models obtain a more accurate forecast of future returns, being closer to the true return values (in most cases) than the frequentist ARMA-GARCH implementation.

The MSMCS Bayesian ARMA-GARCH framework combines the Bayesian ARMA-GARCH's ability to forecast returns accurately—by incorporating prior knowledge and updating beliefs with new information—with MSMCS's efficient sampling of complete distributions, including the tails. This combined framework achieves the first two thesis objectives:

- **Objective one: asset returns forecasting.** Develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

- **Objective two: risk management.** Develop new methodologies which better predict the probability and impact of tail events.

Publication

The proposed MSMCS algorithm has been published:

Millar, Robert, Li, Hui, and Li, Jinglai. "Multicanonical sequential Monte Carlo sampler for uncertainty quantification". Reliability Engineering & System Safety 237 (2023).

Chapter 9

Optimal Portfolio Allocation

In the preceding chapters of this thesis, we systematically analysed methods for (1) forecasting asset returns and volatility under uncertainty and (2) measuring financial risk, specifically tail risk. Future asset returns can be forecast to account for uncertainty, by using methods like Bayesian ARMA-GARCH. Tail risk can be captured through the use of risk measures (like CVaR) by applying MSMCS. Given these insights, one question remains: How can investors achieve an optimal portfolio allocation in uncertain markets?

9.1 Problem set-up

Traditionally employed in Engineering, reliability-based design optimisation (RBDO) incorporates reliability measures, often probabilistic, directly into the optimisation process, as either the objective function or the constraint. By adapting RBDO approaches for financial optimisation, the inherent uncertainty and tail risk in financial modelling can be captured within the optimisation process. We propose using CVaR as the risk measure within the objective of the optimisation problem.

In Section 5.2.4, we found that CVaR operates under stochastic dominance and has a positive relationship to expected return (see section 5.2.4)—features we can

exploit in our optimisation problem design.

In order for an investor to achieve a certain level of expected return, the associated CVaR must be considered—where the most desirable portfolio, achieves the expected return requirement with the lowest possible CVaR. Noting that, to reduce the portfolio's CVaR further would necessitate a lower expected return. On this basis, we establish our optimisation problem as follows.

The optimisation problem has a minimum expected return requirement, set as the constraint, and an objective to minimise the CVaR. We seek to find the optimal allocation of portfolio weights $\mathbf{w}$. Formally, the problem is defined as follows:

$$\min_{\mathbf{w}} F(\mathbf{w}) := \mathrm{CVaR}_{\alpha}[f(\mathbf{w}, \mathbf{Z})] \tag{9.1.1a}$$

$$\text{s.t.} \quad R(\mathbf{w}) := \mathbb{E}_{\mathbf{Z}}[f(\mathbf{w}, \mathbf{Z})] \geqslant r^{\min} \tag{9.1.1b}$$

$$0 \leqslant w_i \leqslant 1,\ i = 1, ..., N, \quad \sum_{i=1}^{N} w_i \leqslant 1. \tag{9.1.1c}$$

9.2 Portfolio optimisation methods

Portfolio optimisation utilising CVaR is not new—various algorithms exist to obtain an optimal solution under such a framework, where CVaR is either in the objective or constraint(s).

9.2.1 Linear programming

One of the most commonly referenced papers in this field is by Rockafellar and Uryasev (2002). The paper aims to minimise CVaR while ensuring a certain level of return. The paper demonstrates that if the reward and loss functions are linear and accessible (not black-box), then the problem can be solved as a classic linear programming (LP) problem, where basic MC simulations are used to generate samples

representing the risk factors from known probability distributions.

Rockafellar and Uryasev (2000) seeks to maximise expected returns under conditional value-at-risk (CVaR) constraints. The paper calculates VaR using either (1) a linear approximation of portfolio risk and assuming joint Normal distributions of market variables, or (2) a Monte Carlo simulation-based method when the portfolio contains non-linear assets. The proposed algorithm considers different expected returns and their associated risks to generate an efficient frontier, from which the optimal portfolio allocation is obtained.

Hochreiter (2007) propose using an evolutionary exploration with LP to consider a finite set of scenarios with different probabilities, to which an evolutionary algorithm is applied and improving solutions that meet the CVaR constraint are accepted, as determined using standard LP.

The LP approaches all rely on the assumption of linearity; an oversimplification. Markets often exhibit non-linear behaviour due to their market dynamics and complex financial instruments. As such, LP methods have the potential to misrepresent the risk and return dynamic.

9.2.2 Non-linear and other traditional programming

Gaivoronski and Pflug (2005) published a portfolio optimisation approach in the non-linear setting, which aims to establish an efficient frontier by minimising VaR for different values of minimal return—from which the optimal allocation is estimated using standard non-linear programming (NLP) methods. The VaR is determined using a smoothed VaR function based on historical VaR data to reduce the effect of outliers.

Other traditional approaches include: Ghaoui et al. (2003) propose a conic optimisation approach. Pirvu (2007) propose a sequential quadratic programming

approach. Finally, Wozabal (2012) propose converting the VaR into difference of convex functions. Whereby the distribution of the underlying random variables becomes discrete, so the branch-and-bound method can be applied.

Ahmadi-Javid and Fallah-Tafti (2019) propose using the primal-dual algorithm to solve investment optimisation problems, where Newton's method is applied to the KKT equations. Such gradient-based methods assume differentiability of the return function, which is not always viable, as return distributions can be discontinuous or exhibit jumps due to market events. As such, non-gradient-based methods are desirable.

9.2.3 Heurstic and simulation-based methods

In their study, Gilli and Këllezi (2002) present a heuristic technique that uses Threshold Accepting, a deterministic form of Simulated Annealing, to search through the solution space. Specifically, the paper aims to maximise returns under a VaR or CVaR constraint by conducting a local search where the generated samples can escape local minima by accepting solutions not worse than a specified threshold. The threshold is reduced at each iteration until it reaches zero, at which point only improving solutions are accepted. (C)VaR is determined using modelled future returns, which are randomly drawn from the empirical distribution of past returns. The work was extended to use actual data in Gilli et al. (2006).

Nguyen et al. (2011) propose a Cross-Entropy method for finding an optimal portfolio with VaR constraints. The paper applies CE to improve the design variable quality by encouraging sampling in the optimal return-maximising region. VaR is determined for each proposed variable and then an acceptance threshold is applied, in which the objective function is set to zero if the VaR violates this threshold.

9.3 Motivation for new approach

The assumptions of linearity, differentiability, or accessible objective and constraint functions underlie the optimisation algorithms set out above. In many financial settings, the return functions are not linear, and cannot be accessed directly, making these approaches infeasible. Additionally, none of these methods allows for the incorporation of uncertainty into the framework; a central objective of this thesis. Therefore, an alternative approach to portfolio optimisation is needed; one that avoids any unrealistic assumptions imposed on the objective and constraint functions, can work with black-box functions, and incorporates uncertainty into the procedure.

Some progress towards this has been made, nicely summarised in the literature reviews by Kalayci et al. (2019) and Xidonas et al. (2020). The reviews compared various portfolio optimisation algorithms—concluding that advanced deterministic models, such as robust optimisation and stochastic programming, offer a more realistic approach by considering uncertainty and allowing non-normal return distributions. *Robust optimisation* focuses on creating portfolios that perform well under worst-case scenarios, safeguarding against extreme market movements. *Stochastic programming* utilises dynamic portfolio adjustments to adapt to changing market conditions.

Through the optimisation problem set-up we established in Section 9.1, our proposed framework can be considered robust—as it considers worst-case scenarios through the CVaR objective. To solve our optimisation problem, we propose the application of a stochastic programming procedure. Stochastic, also called *search-based*, algorithms have the added advantage of being able to handle the multiple local extrema or discontinuities that return and risk functions often display. Stochastic

algorithms do not impose assumptions on statistical properties of optimisation functions, including allowing for black-box functions; and—as we will see—stochastic algorithms can account for uncertainty. The class of robust stochastic optimisation algorithms we utilise are *Bayesian Optimisation*.

9.4 Bayesian Optimisation

With the increasing complexity of financial markets and instruments, portfolio optimisation faces challenges, especially when dealing with noisy and expensive black-box objective and constraint functions. Traditional optimisation methods are inadequate for such problems. Bayesian optimisation (BO) methods have gained attention for their effectiveness in handling noisy, expensive and black-box functions. The BO procedure is detailed in the next chapter.

BO methods are applicable to scenarios where the objective function does not have a closed-form expression, but noisy evaluations can be obtained at sampled points (Brochu et al., 2010). The BO method is highly beneficial when the function evaluations are costly, when derivatives are inaccessible, or when dealing with non-convex problems—all of which are characteristics of portfolio optimisation problems. The effectiveness of Bayesian optimisation hinges on two fundamental components (Brochu et al., 2010): firstly, it leverages the entire history of samples to construct a posterior distribution over the unknown objective function; secondly, it employs an acquisition function to balance exploration and exploitation in selecting subsequent sampling points. Consequently, Bayesian optimisation techniques are among the most efficient methods in terms of minimising the number of function evaluations required.

Cakmak et al. (2020) propose a BO algorithm for the unconstrained optimisation of risk measures—specifically, VaR and CVaR—of black-box expensive-to-evaluate

functions, with randomness induced by an environmental random variable. The paper proposes modelling the underlying function f as a GP and applying a BO method which jointly selects both a portfolio weight $\mathbf{w}$ and realisation of the environmental variable $\mathbf{Z}$, using a one-step look-ahead approach based on the knowledge gradient acquisition function.

Furthermore, Nguyen et al. propose two alternate BO approaches for optimising VaR (Nguyen et al., 2021b) and CVaR (Nguyen et al., 2021a), which provide certain desirable computational and theoretical properties—further explored in the next chapter. More recent work includes Daulton et al. (2022a), which aims to tackle multivariate value-at-risk problems and Picheny et al. (2022), which studies the situation that the distribution of $\mathbf{Z}$ is unavailable.

The aforementioned methods are designed for unconstrained minimisation of risk measures. Designing a BO method that seeks to minimise the risk measure under a minimum expected return constraint is highly desirable. A popular class of BO methods for general constrained optimisation problems incorporate the constraints into the acquisition function design (Gramacy and Lee, 2011; Gardner et al., 2014; Gelbart et al., 2014).

More recent advances include Lam and Willcox (2017), Letham et al. (2019) and Eriksson and Poloczek (2021), among others. Whilst these methods are effective, they often require frequent evaluation of the risk measure functions, which is unsuitable for complex allocation problems. Hence, we are motivated to develop a BO algorithm for optimal portfolio allocation, where we seek to minimise a risk measure under a minimum return requirement or constraint.

Chapter 10

Bayesian Optimisation for Portfolio Allocation

This chapter introduces Gaussian Processes (GPs) for modelling complex, non-linear relationships in data, effectively incorporating uncertainty through probabilistic predictions. GPs are the backbone of GP-based regression, which enables flexible, non-parametric modelling without assuming a specific form for the underlying function. Bayesian Optimisation (BO)—which utilises GPs—is employed to solve our portfolio optimisation problem, seeking to minimise conditional value-at-risk (CVaR) under a minimum expected return constraint.

By applying BO for this task, we aim to harness the strengths of GPs in capturing uncertainty and the efficiency of BO in navigating complex solution spaces—aligning with our overall thesis motivation and objectives. We propose several adaptations to existing BO procedures, designed to take advantage of two key properties of portfolio allocation problems: 1) the expected return constraint functions are much cheaper to evaluate than the objective function, i.e., the risk measures; 2) the expected return constraints are typically *active* – namely, the optimal solution lies on the boundary of the feasible region defined by the constraints. Both statements are justified within

this chapter.

We introduce a two-stage BO adaptation, which reduces the number of full-function evaluations needed to find an optimal solution, significantly reducing the algorithm's computational cost. Only samples that meet certain criteria are fully evaluated in the second stage; this differs from a cascade-based BO approach (see e.g., Kusakawa et al. (2022)) where all samples in the first stage are used in the second, regardless of their feasibility or promise.

By taking advantage of our problem set-up, we propose a new acquisition function that encourages more samples to be generated in the near-optimal region, improving the algorithm's performance. To take advantage of parallel computing, we detail how our proposed BO algorithms can be adapted for batch implementation.

The proposed BO algorithms are highly effective for solving constrained portfolio allocation problems, outperforming the current best constrained BO approaches with a lower computational cost and, crucially, faster convergence to an optimal solution—as demonstrated in later numerical examples. These improvements are achieved by combining a new acquisition function, a two-stage procedure and the potential for parallel batch implementation. We begin by introducing GP and GP-regression, the backbone of Bayesian Optimisation.

10.1 Gaussian Process

A Gaussian Process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP model provides a framework for conducting non-parametric regression in the Bayesian fashion. As defined by Rasmussen (2003), the GP model for a function $g(x)$ can be written as:

$$g(x) \sim GP(\mu(x), k(x, x')), \tag{10.1.1}$$

where the mean function $\mu(x)$ and covariance function $k(x, x')$ are defined for any pair of input points $x, x' \in \mathbb{R}^d$, by:

$$\begin{aligned} \mu(x) &= \mathbb{E}[g(x)], \\ k(x, x') &= \mathbb{E}[(g(x) - \mu(x))(g(x') - \mu(x'))]. \end{aligned} \tag{10.1.2}$$

Therefore, given a set of points $\mathbf{x} = \{x_1, ..., x_T\}$ with corresponding function values $g(\mathbf{x}) = \{g(x_1), ..., g(x_T)\}$, the marginal distribution is a multivariate Gaussian distribution:

$$g(\mathbf{x}) \sim N(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x})), \tag{10.1.3}$$

with mean vector $\mu(\mathbf{x})$ and positive-definite covariance matrix & function $\Sigma = k(\mathbf{x}, \mathbf{x})$.

The specification of the covariance function $k(\mathbf{x}, \mathbf{x})$—or *kernel function*—allows one to set prior information over the function $g(\mathbf{x})$. In our procedure, we chose to use the Matérn 5/2 kernel due to its balance between flexibility and accuracy in the function representation. This is particularly useful since our underlying process is unlikely to be perfectly smooth. The Matérn 5/2 kernel is defined for two points x and x' as:

$$k_{Matern52}(x, x') = \sigma^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right), \text{where} \tag{10.1.4}$$

$$r = \sqrt{\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{l_i^2}}. \tag{10.1.5}$$

r is an adapted measure of the Euclidean distance between x and x', l_i are the length scales, and σ^2 is the variance parameter. Refer to Rasmussen (2003) for further details on the GP design.

Gaussian Processes are used to model complex, often non-linear relationships in

data, accounting for uncertainty by providing both predictions and their associated confidence levels through a probabilistic framework.

10.1.1 GP-based regression

GP-based regression leverages the principles of GPs for inferring complex relationships within data. GP-based regression aims to predict outcomes by constructing a probabilistic model, where the predictions at any set of input points are Gaussian distributed, providing both an estimate and uncertainty measure for each prediction.

The GP-based regression proceeds as follows. Based on a set of input points $\mathbf{x}$, referred to as the training set, GP-based regression seeks to evaluate a new proposed design point $\widehat{x}$. To start, the joint Gaussianity of all finite subsets implies:

$$\begin{bmatrix} g(\mathbf{x}) \\ g(\widehat{x}) \end{bmatrix} = N\left(\begin{bmatrix} \mu(\mathbf{x}) \\ \mu(\widehat{x}) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x},\mathbf{x}) & k(\mathbf{x},\widehat{x}) \\ k(\widehat{x},\mathbf{x}) & k(\widehat{x},\widehat{x}) \end{bmatrix} \right) \tag{10.1.6}$$

From this, it is possible to obtain the posterior distribution of $g(\widehat{x})$ conditional on the training data set $D = \{\mathbf{x},\, g(\mathbf{x})\}$. The posterior distribution is Gaussian with $N(\tilde{\mu}(\widehat{x}), \tilde{\Sigma}(\widehat{x}))$:

$$\begin{aligned} \tilde{\mu}(\widehat{x}) &= \mu(\widehat{x}) + k(\widehat{x},\mathbf{x})k(\mathbf{x},\mathbf{x})^{-1}(g(X) - \mu(\mathbf{x})), \\ \tilde{\Sigma}(\widehat{x}) &= k(\widehat{x},\widehat{x}) - k(\widehat{x},\mathbf{x})k(\mathbf{x},\mathbf{x})^{-1}k(\mathbf{x},\widehat{x}). \end{aligned} \tag{10.1.7}$$

The posterior mean, $\tilde{\mu}(\widehat{x})$, updates our prediction for $g(\widehat{x})$, incorporating the influence of the training data by adjusting the prior mean with a weighted difference of observed and expected outputs. The posterior covariance, $\tilde{\Sigma}(\widehat{x})$, quantifies the uncertainty of this prediction, reducing the prior variance based on the information gained from the training data. These computations are central to GP regression, enabling predictions informed by the data and quantifying uncertainty.

Having introduced the GP-regression, we proceed with the presentation of Bayesian Optimisation; this chapter first explains Bayesian Optimisation in the general setting with objective $g(\mathbf{x})$ and some constraints. Then, it presents Bayesian Optimisation for portfolio allocation problems and introduces our proposed adaptations.

10.2 General Bayesian Optimisation

Bayesian Optimisation (BO) is a probabilistic framework for optimising black-box functions based on the GP model (Močkus, 1975). In the unconstrained setting, BO sequentially evaluates the objective function at selected points, from which a GP model of the target function is constructed. The design point(s) are selected by maximising an acquisition function—detailed in the next subsection—which quantifies a desired trade-off between the exploration and exploitation of the GP model.

This fundamental trade-off exists in finite resource problems, where the decision-making process can be oriented towards two main strategies: exploitation or exploration. Exploitation involves making the best decision based on current information, focusing on known and proven methods or areas. In contrast, exploration is about gathering more information by venturing into unknown or less understood areas. This trade-off is crucial in optimising the performance of GP models, where balancing these two approaches can lead to more effective and informed decision-making.s

The standard BO procedure for unconstrained problems is given in Alg. 16.

Algorithm 16 Bayesian Optimisation

Input: Objective function $g(x)$, acquisition function $a(x, \tilde{g})$, initial design for training data set D_0, stopping criteria
Initialize the training data set D_0 using an initial design
Let $t = 0$
while stopping criteria not met **do**
    Let $t = t + 1$
    Construct a Gaussian Process (GP) model $\tilde{g}_{t-1}$ using D_{t-1}
    Find $x_t = \arg\max_x a(x, \tilde{g}_{t-1})$
    Update $D_t = D_{t-1} \cup \{x_t, g(x_t)\}$
end while
Return: The point x_t that globally minimises $g(x)$ based on the GP model

10.2.1 General BO acquisition functions

Bayesian Optimisation utilises various acquisition functions to guide the search process. Three popular acquisition functions are Expected Improvement, Probability of Improvement, and Upper Confidence Bound.

Expected Improvement

The Expected Improvement (EI) function maximises the expected improvement over the current best-known point. It balances exploration and exploitation effectively and has strong theoretical and empirical support, as demonstrated in the works by Turner et al. (2020) and Snoek et al. (2012). However, it is less effective in high-dimensional spaces due to the curse of dimensionality. The Expected Improvement for a proposed point x is defined as:

$$EI(x) = \mathbb{E}[\max\{0, g(x^+) - g(x)\}], \tag{10.2.1}$$

where $g(x^+)$ is the value of the best sample so far, and the expectation is taken with respect to the posterior distribution of $g(x)$.

Probability of Improvement

Probability of Improvement (PI) is a probabilistic adaptation of EI, which maximises the probability that a new data point will improve on the current best observation (Turner et al., 2020). PI uses incremental improvements, tending to propose points with small but certain improvements, potentially missing out on areas with higher uncertainty but larger potential gains. The Probability of Improvement for a proposed point x is defined as:

$$PI(x) = \mathbb{P}(g(x) < g(x^+)), \tag{10.2.2}$$

where $g(x^+)$ is the value of the best sample so far, and $\mathbb{P}$ denotes probability.

Upper Confidence Bound

The Upper Confidence Bound (UCB) function is the sum of the prediction and a term proportional to the uncertainty (Brochu et al., 2010). UCB provides explicit control over the exploration-exploitation trade-off using a parameter denoted by κ, which is problem-specific and often challenging to select. The Upper Confidence Bound for a proposed point x is defined as:

$$UCB(x) = \mu(x) + \kappa\sigma(x), \tag{10.2.3}$$

where $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation of the posterior distribution at x.

10.2.2 General constrained BO

This section presents the BO method for optimisation problems with inequality constraints, largely following Gardner et al. (2014) and Gelbart et al. (2014)—based on the following constrained optimisation problem:

$$\min_{x} \; g(x) \text{ s.t. } c_k(x) \leqslant 0, k = 1, ..., K. \tag{10.2.4}$$

To utilise the BO method to solve Eq.(10.2.4), it is necessary to model all the constraint functions $c_k(x)$ as separate GPs. Specifically, the GP model for the k^{th} constraint function $c_k(x)$ is obtained from the constraint training set $C^k = \{(x_1, c_k(x_1)), ..., (x_m, c_k(x_m))\}$. For each design point, every constraint function must be evaluated to improve the GP $\tilde{c}_k(x)$ for the corresponding constraint function $c_k(x)$. In the BO procedure, to assess the objective and constraints for each proposed design point, the acquisition function should incorporate the constraints.

As per Gardner et al. (2014), let x^+ be the current best evaluated point, i.e., the point with the smallest $g(x^+)$ value in the current training set. Then, conditional on this training set C^k, define the probability of feasibility (PF) to be:

$$\text{PF}(x) = \mathbb{P}(\tilde{c}_1(x) \leqslant 0, \, \tilde{c}_2(x) \leqslant 0, \, ..., \, \tilde{c}_K(x) \leqslant 0). \tag{10.2.5}$$

This represents the probability that a candidate point x satisfies all the constraints. Then, by conditional independence of the constraints given x:

$$\text{PF}(x) = \prod_{k=1}^{K} \mathbb{P}(\tilde{c}_k(x) \leqslant 0). \tag{10.2.6}$$

Incorporating this with the EI acquisition function obtains a new acquisition func-

tion:

$$a_{\text{CW-EI}}(\mathbf{x}) = \text{EI}(x)\text{PF}(x),, \tag{10.2.7}$$

which, as per Gardner et al. (2014), is referred to as the 'constraint-weighted expected improvement' (CW-EI) acquisition function. This acquisition function considers the expected improvement in the objective and the probability of feasibility for the constraints.

The constrained BO algorithm proceeds largely the same as the unconstrained version (Alg. 16), except for the following two main differences: (1) the constrained acquisition function in Eq. (10.2.7) is used to select the new design points; (2) for each design point, both the objective and constraint functions are evaluated. We hereafter refer to this constrained BO method as 'CW-EI BO'.

10.3 Bayesian optimisation for portfolio allocation

Gardner et al. (2014) and Gelbart et al. (2014) established a constrained Bayesian Optimisation framework to account for uncertainty caused by an environmental random variable (i.e., $\mathbf{Z}$). This framework has been applied in the general setting but, to the best of our knowledge, never for portfolio optimisation. Our proposed BO procedure builds upon this work but includes several adaptations to enhance performance and leverage the specific problem structure related to portfolio allocation.

It is possible to apply a class of BO approaches (see e.g., Fröhlich et al. (2020); Cakmak et al. (2020); Daulton et al. (2022b)) which model the function f as a single GP for a fixed environmental variable $\mathbf{z}$ during the optimisation procedure, where the variable $\mathbf{Z}$ is only random during implementation. Whilst this may be appropriate for some optimisation problems, it is not for portfolio allocation problems where

the full range of $\mathbf{Z}$ must be considered within the optimisation procedure to ensure uncertainty and tail risk are adequately accounted for.

The constraint-weighted expected improvement BO procedure could be applied directly to the portfolio optimisation problem—as set up in Section 9.1. CW-EI would model the CVaR objective and expected return constraint as separate GPs and the then standard procedure is followed (Alg. 16). Within this, for each proposed portfolio allocation $\mathbf{w}$, the expected return and CVaR with respect to $f(\mathbf{w}, \mathbf{Z})$ must be obtained. This could be achieved through a standard MC simulation, or a more advanced technique (i.e., the proposed MSMCS algorithm).

The work in Chapter 6, demonstrated that a small MC sample size can be used to obtain an expectation from $f(\mathbf{w}, \mathbf{Z})$, however, to obtain an accurate estimate of the tail risk, or CVaR, a very large number of samples is required. As such, the computational cost of evaluating the CVaR objective—in the BO procedure—is much higher than evaluating the expected return constraint[1]. This fact is central to our proposed BO algorithm.

Our proposed BO algorithm is an adaptation of the CW-EI BO method, motivated by our observation that the computational cost of evaluating CVaR is significantly higher than the expected return, so the computational efficiency of BO can be enhanced by reducing the number of CVaR evaluations. In addition, the proposed BO algorithm is specifically designed to take advantage of key features unique to portfolio allocation problems—which we detail now.

10.3.1 Activeness of the minimum expected return constraint

To ease understanding, we restate the VaR and CVaR definitions.

Definition 10.3.1 *The VaR of $f(\mathbf{w}, \mathbf{Z})$ at a specified risk level $\alpha \in (0, 1)$, denoted*

[1]For example, in our later numerical examples (in Sect. 10.6.2), the cost for evaluating the expected return is around 1% of the cost for evaluating CVaR.

as $v_f(\mathbf{w};\alpha)$, is defined as the threshold value ω such that the probability of the loss exceeding ω is at most $(1-\alpha)$. Formally, VaR is defined as:

$$v_f(\mathbf{w};\alpha) = \inf\{\omega : \mathbb{P}(f(\mathbf{w},\mathbf{Z}) \leqslant -\omega) \leqslant 1-\alpha\}. \tag{10.3.2}$$

Definition 10.3.3 *The CVaR of $f(\mathbf{w},\mathbf{Z})$ at a specified risk level $\alpha \in (0,1)$ is defined as the expected loss, assuming that the loss is worse than the VaR threshold. It represents the average of the worst-case losses. Formally, CVaR is defined as:*

$$CVaR_\alpha[f(\mathbf{w},\mathbf{Z})] = -\mathbb{E}[f(\mathbf{w},\mathbf{Z})|f(\mathbf{w},\mathbf{Z}) \leqslant -vf(\mathbf{w};\alpha)]. \tag{10.3.4}$$

It is important to note that $f(\mathbf{w},\mathbf{z}) \leqslant 0$ indicates losses, whereas VaR and CVaR are statements about the losses, so $v_f(\mathbf{w};\alpha) \geqslant 0$ and $\text{CVaR}_\alpha[f(\mathbf{w},\mathbf{Z})] \geqslant 0$ represent negative returns, or losses. We introduce and prove several assumptions in relation to the return function $f(\mathbf{w},\mathbf{z})$ and the distribution of $\mathbf{Z}$.

Assumption 1. *(a) $f(\mathbf{w},\mathbf{z})$ is a continuous function of $\mathbf{w}$ for any fixed $\mathbf{z}$; (b) $f(\mathbf{0},\mathbf{z}) \equiv 0$; (c) for a given $\mathbf{w} \in \mathbb{W}$ and any fixed $\mathbf{z}$, if $f(\mathbf{w},\mathbf{z}) \leqslant 0$, $f(\rho\mathbf{w},\mathbf{z})$ is a decreasing function of $\rho \in [0,1]$; (d) there exists $\alpha \in (0,1)$ such that $v_f(\mathbf{w};\alpha) \geqslant 0$ $\forall$ $\mathbf{w} \in \mathbb{W}$.*

- Assumption 1.(a) ensures that small changes in portfolio allocation do not lead to abrupt or unpredictable changes in outcomes; a reasonable expectation in most financial models.

- Assumption 1.(b) is straightforward; an absence of investment will result in a neutral (zero) financial return.

- Assumption 1.(c) implies that, if a chosen portfolio allocation results in a loss for a certain scenario, this loss does not increase if the total capital is

proportionally reduced[2]; this reflects the intuitive notion that if investing a certain amount leads to a loss, investing less should not lead to a greater loss.

- Assumption 1.(d) implies that there always exists a choice of $\alpha \in (0, 1)$ so that no matter the allocation ($\mathbf{w} \in \mathbb{W}$), $v_f(\mathbf{w}; \alpha)$ is positive, i.e., a loss. In simpler terms, no matter the allocation, there always exists some level of risk (represented by α), which can be chosen to ensure there is always some risk of loss (as indicated by VaR). This is important, as it allows us—through the appropriate choice of α—to just consider the loss scenarios when evaluating the associated CVaR.

From this, we obtain the following theorem:

Theorem 10.3.5 *If function $f(\mathbf{w}, \mathbf{Z})$ and distribution $p_{\mathbf{z}}(\cdot)$ satisify Assumptions 1, α is chosen such that $v_f(\mathbf{w}, \alpha) \geqslant 0 \; \forall \; w \in \mathbb{W}$, and solutions to the constrained optimisation problem* (9.1.1) *exist, then there must exist a solution to problem* (9.1.1)*, denoted as $\mathbf{w}^*$, such that $R(\mathbf{w}^*) = r^{min}$.*

Before we prove this theorem, we interpret it. Simply put, Theorem 1 states that, under some reasonable assumptions, the constraint (9.1.1b) is active for at least one solution. The result is rather intuitive, as it infers that a higher expected return can only be obtained by increasing risk exposure and, as such, the CVaR. The optimal solution to our problem will likely arise from an active constraint, where the minimum expected return requirement limits our ability to reduce the CVaR further. This aligns with our earlier analysis in Section 5.2.4 on CVaR's properties of stochastic dominance and its relationship to returns.

[2]For clarity, as ρ goes from 0 to 1, f goes from $f(0, \mathbf{z}) \equiv 0$ to $f(\mathbf{w}, \mathbf{z})$. As $f(\mathbf{w}, \mathbf{z}) \leqslant 0$, the function value $f(\rho\mathbf{w}, \mathbf{z})$ gets more negative, so f is a decreasing function w.r.t. $\rho \in [0, 1]$.

Proof. First, assume that $\mathbf{w}'$ is a solution to the constrained optimisation problem (9.1.1). It follows directly that $R(\mathbf{w}') \geqslant r^{\min}$. Obviously if $R(\mathbf{w}') = r^{\min}$, the theorem holds.

Now consider the case that $R(\mathbf{w}') > r^{\min}$, i.e., it does not lie on the boundary of the feasible region. From Assumption 1.(a), $R(\mathbf{w})$ is a continuous function of $\mathbf{w}$ in $\mathbb{W}$. Next define a function

$$h(\rho) = R(\rho\mathbf{w}')$$

for $\rho \in [0, 1]$. As $R(\mathbf{w})$ is a continuous function in $\mathbb{W}$, $h(\rho)$ is a continuous function too.

From Assumption 1.(b), we know that $h(0) = 0$, and therefore,

$$h(0) = 0 < r^{\min} < h(1) = R(\mathbf{w}')$$

According to the intermediate value theorem on continuous functions, there exists some $\rho^* \in (0, 1)$ such that $h(\rho^*) = R(\rho^*\mathbf{w}') = r^{\min}$. Let $w^* = \rho^*\mathbf{w}'$ denote this point, which lies on the constraint boundary—we wish to compare $F(\mathbf{w}^*)$ and $F(\mathbf{w}')$, i.e., the CVaR values at these two points for a fixed α.

From the Theorem's assumption, we have $v_f(\mathbf{w}', \alpha) \geqslant 0$ and $v_f(\mathbf{w}^*, \alpha) \geqslant 0$. From Assumption 1.(c), we know that for any $\mathbf{z}$, if $f(\mathbf{w}', \mathbf{z}) \leqslant 0$, then $f(\mathbf{w}', \mathbf{z}) \leqslant f(\mathbf{w}^*, \mathbf{z}) \leqslant 0$.

It follows that for any $\mathbf{z} \in \{\mathbf{z} | f(\mathbf{w}', \mathbf{z}) \leqslant -v_f(\mathbf{w}', \alpha)\}$, we have

$$f(\mathbf{w}', \mathbf{z}) \leqslant f(\mathbf{w}^*, \mathbf{z}) \leqslant -v_f(\mathbf{w}^*, \alpha) \leqslant 0.$$

As such, we can derive $v_f(\mathbf{w}^*, \alpha) \leqslant v_f(\mathbf{w}', \alpha)$, and obtain,

$$\begin{aligned}
\mathrm{CVaR}_\alpha[f(\mathbf{w}^*, \mathbf{Z})] &= -\mathbb{E}[f(\mathbf{w}^*, \mathbf{Z}) | f(\mathbf{w}^*, \mathbf{Z}) \leqslant -v_f(\mathbf{w}^*; \alpha)] && (10.3.6)\\
&\leqslant -\mathbb{E}[f(\mathbf{w}^*, \mathbf{Z}) | f(\mathbf{w}^*, \mathbf{Z}) \leqslant -v_f(\mathbf{w}'; \alpha)] && (10.3.7)\\
&\leqslant -\mathbb{E}[f(\mathbf{w}', \mathbf{Z}) | f(\mathbf{w}', \mathbf{Z}) \leqslant -v_f(\mathbf{w}'; \alpha)] && (10.3.8)\\
&= \mathrm{CVaR}_\alpha[f(\mathbf{w}', \mathbf{Z})], && (10.3.9)
\end{aligned}$$

Therefore, $\mathbf{x}^*$ is also a minimal solution w.r.t. the objective function and $R(\mathbf{x}^*) = r^{\min}$. The proof is thus complete. □

10.3.2 Two-Stage point selection

Theorem 1 proves that under some reasonable assumptions, the constraint (9.1.1b) is active for at least one solution—where the minimum expected return requirement limits the ability to reduce the CVaR further. We seek to exploit this feature.

Based solely on the expected return of a proposed portfolio weight $\mathbf{w}$, one may decide not to evaluate the CVaR objective function in the following two situations. Firstly, if the expected return is lower than the minimum constraint threshold, then the proposed design point is not feasible and as such, the CVaR function does not need to be evaluated. Secondly, if the expected return is too high (i.e., not approximately active), the corresponding CVaR is likely far from optimal—as per Thm. 1—so the objective need not be evaluated. To formalise *'too high'*, we propose the introduction of a maximum expected return parameter, denoted by $r^{\max}$, set on the basis that those points with expected returns higher than this value are highly unlikely to be optimal.

Based on these observations and $r^{\max}$, we propose a two-stage point selection procedure. The first stage selects a design point based on the acquisition function.

The second stage calculates the expected return and only if the expected return satisfies

$$r^{\min} \leqslant R(\mathbf{w}) = \mathbb{E}_{\mathbf{Z}}[f(\mathbf{w}, \mathbf{Z})] \leqslant r^{\max}, \tag{10.3.10}$$

the objective function is evaluated, and the constraint and objective GPs are updated. If Eq. (10.3.10) is not satisfied, the proposed point is rejected and the objective function is not evaluated. To ensure the same point is not reproposed, the expected return constraint GP is updated (but not the objective function GP).

This two-stage (2S) adaptation has the advantage of only fully evaluating feasible and (approximately) active points. As such, it reduces the number of evaluations of the expensive-to-evaluate CVaR objective. Through this process, the algorithm obtains two training sets, one for the CVaR objective and one for the expected return constraint, with the former being a subset of the latter.

10.3.3 New acquisition function

Using the two-stage selection procedure ('2S'), significantly more evaluations of the expected return constraint will occur compared to the CVaR objective. As such, the GP for the constraint will be more accurate than the objective function's. The CW-EI acquisition function would be highly effective at proposing feasible points—due to the quality of the constraint GP—but may be poor at proposing points with low CVaR—due to the lower quality of the objective GP. We propose a new acquisition function to address this issue.

As the CW-EI acquisition function only accounts for the feasibility of the constraint, we want to incorporate the activeness as well. Let $\widetilde{R}(\mathbf{w})$ be a GP model of the expected return $R(\mathbf{w})$, we define

$$\mathrm{PF}(\mathbf{w}) = \mathbb{P}(r^{\min} \leqslant \widetilde{R}(\mathbf{w}) \leqslant r^{\max}), \tag{10.3.11}$$

which is the probability that a design point $\mathbf{w}$ is both feasible and approximately active. Therefore, given $\mathbf{w}$, we have

$$\begin{aligned} \mathrm{PF}(\mathbf{w}) &= \mathrm{PF}_{\min}(\mathbf{w}) \times \mathrm{PF}_{\max}(\mathbf{w}), \text{ where} \\ \mathrm{PF}_{\min}(\mathbf{w}) &= \mathbb{P}(\widetilde{R}(\mathbf{w}) \geqslant r^{\min}) \\ \mathrm{PF}_{\max}(\mathbf{w}) &= \mathbb{P}(\widetilde{R}(\mathbf{w}) \leqslant r^{\max}). \end{aligned} \tag{10.3.12}$$

Combining Eq. (10.3.12) with the Expected Improvement we obtain:

$$a_{\text{ACW-EI}}(\mathbf{w}) = \mathrm{EI}(\mathbf{w})\mathrm{PF}_{\min}(\mathbf{w})\mathrm{PF}_{\max}(\mathbf{w}), \tag{10.3.13}$$

which is hereafter referred to as the 'active constraint-weighted expected improvement' (ACW-EI) acquisition function. Note that this acquisition function depends on both the GP models for CVaR and the expected return—denoted by $a_{\text{CW-EI}}(\mathbf{w}, \widetilde{F}, \widetilde{R})$.

The new term $\mathrm{PF}_{\max}$ in the acquisition function encourages proposed points to be approximately active, which, by proxy, increases the likelihood they are near-optimal with respect to the objective function. The choice of $r^{\max}$ is explored through our numerical examples.

The inclusion of this parameter is a crucial aspect of our proposed BO algorithms. Two feasible points with different true CVaRs are likely to have a similar expected improvement (prior to full evaluation) due to the low-quality GP for the objective function and equal probability of feasibility for the constraint (namely, $= 1$). By introducing the new $r^{\max}$ term—based on the more accurate expected return GP—our proposed BO procedure can differentiate between these two points during the selection procedure. The procedure will choose the point with the lower (but feasible) expected return, which, in turn, is more likely optimal.

10.4 The complete BO algorithm

To complete our proposed algorithm, we must discuss the weight constraints:

$$0 \leqslant w_i \leqslant 1,\ i = 1, ..., N, \tag{10.4.1}$$

$$\sum_{i=1}^{N} w_i \leqslant 1, \tag{10.4.2}$$

which will be denoted as $\mathbf{w} \in S$ in what follows. It is possible to deal with these constraints in the same manner as the expected return, i.e., as GP models. However, unlike the expected return constraint, which is probabilistic, the summation constraint is deterministic and easy to evaluate. As such, we impose the constraint during the maximisation of the acquisition function by solving the following constrained maximization problem: $\max_{\mathbf{w}\in S} a_{\text{ACW-EI}}(\mathbf{w})$, which in this work is solved with the barrier method. In our numerical examples (which follow), we assume that any remaining capital obtains zero return. Alternatively, one could assume that the remaining capital gains the risk-free rate (thereby treating it as another potential asset, i.e., Cash).

Finally, by combining the two-stage point selection, the ACW-EI acquisition function, and the constrained acquisition maximisation, we obtain our complete 2S-ACW-EI BO algorithm, detailed in Alg. 17.

10.5 Batch implementation

In most Bayesian optimisation approaches, the acquisition function is used to select a single point for evaluation. The posterior GPs are then updated based on this evaluation, and the process is repeated. This approach is *sequential* in nature, as each point is selected and evaluated one at a time.

Algorithm 17 The 2S-ACW-EI BO algorithm

Input: Initial training data sets D (for the objective) and C (for the constraint), stopping criteria, bounds $r^{\min}$ and $r^{\max}$ for the constraint, search space S, acquisition function $a_{\text{ACW-EI}}$
Initialise the training data sets D and C using an initial design
Let $t = 1$
while stopping criteria not met **do**
    Construct a GP model $\widetilde{F}_{t-1}$ using D
    Construct a GP model $\widetilde{R}_{t-1}$ using C
    Find $\widehat{\mathbf{w}} = \arg\max_{\mathbf{w} \in S} a_{\text{ACW-EI}}(\mathbf{w}, \widetilde{F}_{t-1}, \widetilde{R}_{t-1})$
    Evaluate the constraint $R(\widehat{\mathbf{w}})$
    Update $C = C \cup \{\widehat{\mathbf{w}}, R(\widehat{\mathbf{w}})\}$
    if $r^{\min} \leqslant R(\widehat{\mathbf{w}}) \leqslant r^{\max}$ **then**
        Evaluate the objective $F(\widehat{\mathbf{w}})$
        Update $D = D \cup \{\widehat{\mathbf{w}}, F(\widehat{\mathbf{w}})\}$
        Let $t = t + 1$
    end if
end while
Return: Optimised dataset D for the objective function under constraints

As it is expensive to evaluate the objective function, it may be advantageous to evaluate several points simultaneously, for example by using parallel computers. In this regard, a batch implementation of our proposed BO procedure is desirable, where several design points are proposed using the acquisition function and then evaluated simultaneously in parallel.

In most batch BO methods, the batch of design points is determined sequentially via a given point-selection procedure, from which the objective and constraint functions are evaluated after the whole batch is obtained. To adapt our two-stage method for batch implementation, we propose including the constraint evaluation as part of the batch point-selection procedure; only once the whole batch is obtained are the CVaR objectives evaluated in parallel.

More specifically, the expected return is evaluated for each new proposed point. If the expected return satisfies Eq. (10.3.10), it is added to the batch and the constraint

GP is updated. If the expected return does not satisfy Eq. (10.3.10), the point is not added to the batch, but the GP for the constraint is still updated to ensure that the point is not proposed again. Once a batch has been determined, each point is fully evaluated—knowing that all points are feasible and approximately active. The pseudo-code for our two-stage batch selection is provided in Alg. 18.

The batch approach can be implemented in parallel, so it has a lower computational cost. However, as the GPs are updated less frequently (than in sequential implementation), each sample is proposed based on a less accurate GP compared to at the equivalent stage in the sequential approach. As such, the batch approach requires a greater total number of samples to converge to the optimal solution.

Algorithm 18 Two-stage batch selection

Input: Training set D for the CVaR objective function, training set C for the expected return constraint, batch size b, constraint bounds $r^{\min}$ and $r^{\max}$, selection rule for design points
Initialize batch $B = \emptyset$
Let $i = 0$
while $i < b$ **do**
    Propose a new design point $\widehat{\mathbf{w}}$ based on the prescribed selection rule
    Evaluate the constraint $R(\widehat{\mathbf{w}})$
    if $r^{\min} \leqslant R(\widehat{\mathbf{w}}) \leqslant r^{\max}$ **then**
        Update $B = B \cup \{\widehat{\mathbf{w}}\}$
        Let $i = i + 1$
    end if
    Update $C = C \cup \{\widehat{\mathbf{w}}, R(\widehat{\mathbf{w}})\}$
    Update the Gaussian Process (GP) model for the constraint using C
end while
Return: Batch B of b design points satisfying the constraints

10.6 Numerical experiments

For a proposed portfolio allocation, it is possible to utilise the Bayesian ARMA-GARCH framework with MSMCS to obtain a forecasted expected return and CVaR. The results of such an implementation are reserved for the final example, which

brings together all aspects of our research. First, we demonstrate the performance of the proposed BO algorithm in isolation through four numerical examples. We begin with a simple mathematical example, which enables us to better explain how the proposed BO algorithm drives sampling and, thus, performance. Then, to demonstrate the performance of the BO algorithms against each other, we introduce three portfolio allocation problems, which are simpler to understand than the complete Bayesian ARMA-GARCH MSMCS framework.

For all the numerical examples, our proposed BO algorithms are implemented using *Trieste*, a BO Python package built on TensorFlow. We used the default Matern 52 Kernel within the package, with length scale 1.0 and likelihood variance $1e^{-7}$. The resulting problem is solved using the Efficient Global Optimisation (EGO) method provided by the package.

10.6.1 Mathematical example

Problem set-up

We first consider a simple mathematical example to demonstrate how the design points are selected by the different BO algorithms. Adapted from Gramacy et al. (2016), we apply the BO algorithms to solve the following constrained optimisation problem:

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) :=& - x_1 - x_2 \\ \text{s.t. } c(\mathbf{x}) :=& \frac{3}{2} - x_1 - 2x_2 - \frac{1}{2}\sin(2\pi(x_1^2 - 2x_2)) \geqslant 0. \end{aligned} \tag{10.6.1}$$

Implementation details

The original CW-EI method, ACW-EI (i.e. the new acquisition function without the 2S process), and 2S-ACW-EI are implemented with 10 initial points and a further

50 iterations.

Results

Figure 10.1 shows the design points obtained by each of the three algorithms. The true optimal solution to the problem is $\mathbf{x} = (0.918, 0.540)$, where $f(\mathbf{x}) = 1.458$.

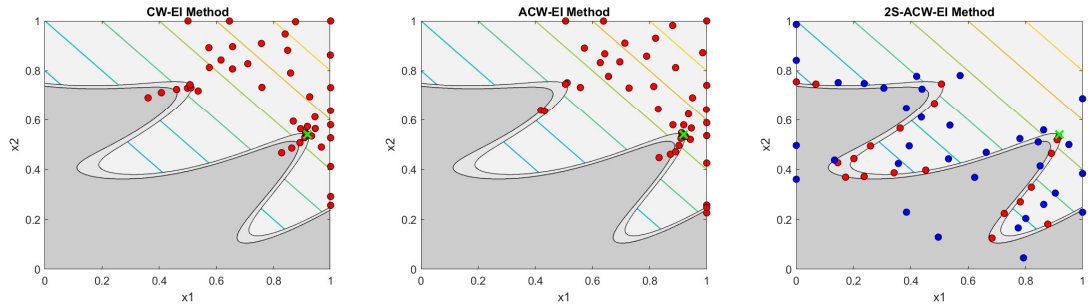


Figure 10.1: Plots showing the optimal solution (green "x") and the design points generated by each method. The figures include the fully evaluated points (red dots) and those for which only the constraint was evaluated (blue dots). The feasible region is dark grey, the active region is light grey and the infeasible region is very light grey. The objective function contours are shown too, going from objectives maximal (blue) to minimal (yellow).

Interpretation of results

The CW-EI and ACW-EIs methods both establish a good GP for the objective function, encouraging samples to be generated in the high objective region, before the GP for the constraint is fully formed. This is reflected in the algorithms' generating a significant number of infeasible samples with high objective values before moving towards the feasible region.

In contrast, in the 2S-ACW-EI method, samples are only fully evaluated if they are in the active region, therefore after a few iterations, the GP for the objective and constraint functions are weak and strong respectively. Thanks to the well-formed constraint GP, the acquisition function prioritises the generation of points in the feasible region, particularly in the active region, before finding those feasible points that are maximised for the objective.

10.6.2 Simple portfolio allocation examples

Introduction

The following three examples are based on an investor seeking to optimally allocate capital to stock or stock options, related to the twenty largest technology companies by market capitalisation listed on American stock exchanges (both the NYSE & Nasdaq). The key data is summarised in Table 10.1, including the historical averages and standard deviations of annual returns; and information related to options, which form the basis of our second and third examples. The data is a snapshot from July $13, 2022$[3].

In this simple example, we assume that future prices can be predicted simply by using a Normal distribution based on the historical means and standard deviations of returns. Clearly, this is a grand simplification of financial market dynamics and forecasting. In the final numerical results, we apply the proposed BO algorithms to our Bayesian ARMA-GARCH MSMCS framework to demonstrate the performance on a more complex and realistic set-up. However, in this example we intentionally use a simple problem set-up, as it allows us to more easily understand the nature of our algorithms and compare their performance, removing factors such as the quality of the forecasting and distribution reconstruction methods.

[3]The stock price data, including historical performance, is from *morningstar.com*; call options price data is from *marketwatch.com* and Greeks' data is from *nasdaq.com*.

Asset i	Company name	Ticker	Stock price ($)	Historic average annual return (%)	Historic return std dev (%)	Strike price ($)	12-month call bid price ($)	Delta	Gamma
1	Apple Inc	AAPL	145.49	34.67	66.63	160	14.60	0.4462	0.0112
2	Microsoft Corp	MSFT	252.72	31.83	42.45	275	21.45	0.4029	0.0068
3	Alphabet Inc	GOOGL	2227.07	29.07	40.46	2,450	222.20	0.4249	0.0007
4	Amazon.com Inc	AMZN	110.40	75.21	196.12	120	15.10	0.4615	0.0119
5	Tesla Inc	TSLA	711.12	116.93	219.27	780	152.90	0.5313	0.0012
6	Meta Platforms Inc	META	163.49	36.96	33.72	180	26.05	0.5160	0.0066
7	Nvidia Corp	NVDA	151.64	59.05	89.51	165	22.35	0.4942	0.0069
8	Broadcom Inc	AVGO	481.73	37.44	26.24	530	37.70	0.3933	0.0032
9	Oracle Corp	ORCL	70.03	36.06	66.03	78	4.75	0.3857	0.0249
10	Cisco Systems Inc	CSCO	42.70	33.10	59.29	47	2.06	0.3896	0.0414
11	Adobe Inc	ADBE	371.94	33.43	52.08	410	39.25	0.4569	0.0038
12	Salesforce Inc	CRM	163.49	34.66	42.79	180	17.80	0.4859	0.0081
13	Intel Corp	INTC	37.21	25.58	51.98	40	3.40	0.4041	0.0393
14	Qualcomm Inc	QCOM	135.64	100.34	469.18	150	15.15	0.4764	0.0091
15	Texas Instruments Inc	TXN	154.29	16.71	36.39	170	10.65	0.4131	0.0110
16	Intuit Inc	INTU	383.31	27.55	42.81	420	44.20	0.4769	0.0035
17	AMD Inc	AMD	77.52	49.76	120.59	85	13.10	0.5152	0.0130
18	IBM Corp	IBM	137.18	8.84	26.60	150	6.60	0.3191	0.0133
19	Paypal Holdings Inc	PYPL	71.36	39.24	47.09	78	11.80	0.5119	0.0144
20	Netflix Inc	NFLX	176.56	72.39	115.07	195	29.85	0.5222	0.0053

Table 10.1: Key financial data as of 13th July 2022

Problem set-up

Take the environment random variable $\mathbf{Z}$ to represent the distribution of forecasted stock prices at a future time (under the Gaussian assumption on historical data). Then in all three examples, we define the return function as:

$$f(\mathbf{w}, \mathbf{z}) = \sum_{i=1}^{20} w_i y_i(z_i), \tag{10.6.2}$$

where y_i is the asset return corresponding to the i-th company, a function of its future stock price z_i. In the three examples, we alter the asset type – namely the function $y_i(z_i)$ varies. In each example, we consider two return constraints, one lower and one higher, to understand the BO algorithm's robustness to this parameter.

Example one. We seek to optimally allocate the investor's capital directly to the twenty stocks, which corresponds to setting $y_i = z_i/\overline{z}_i - 1$ with $\overline{z}_i$ being the stock's purchase price. Example 1 has a constraint for $\alpha = 0.0001$ of $r^{min} =$ (a) 0.45 and (b) 0.55.

Example two. We seek to allocate the investor's capital to European Call options held till expiry, based on the twenty stocks. A European Call option gives the owner the right to purchase the underlying asset for a pre-agreed strike price on a specified future date. Suppose that the present bid price of the call option for the i^{th} stock is b_i and the strike price is K_i, then the asset return is

$$y_i = (\max(0, z_i - K_i) - b_i)/b_i - 1$$

Example 2 has a constraint for $\alpha = 0.0001$ of $r^{min} =$ (a) 4.30 and (b) 4.40.

Example three. We consider European Call options, but where the return is derived from selling the option after six months rather than holding it to maturity. As such, the return depends on the change in the option price. Option prices can be modelled using quadratic functions of the underlying asset returns, realised

through a delta-gamma approximation, that is, a second-order Taylor expansion of the portfolio return (Zymler et al., 2013). Namely, at a particular future time, the associated call option return becomes $y_i = \Delta_i\ \varepsilon + \frac{1}{2}\ \Gamma_i\ \varepsilon^2 - 1$, with $\varepsilon = z_i - \overline{z}_i$. Example 3 has a constraint for $\alpha = 0.0001$ of $r^{min} =$ (a) 1.90 and (b) 2.00.

Results

In all three examples, we applied the three sequential and two batch methods (standard CW-EI batch method and our proposed 2S batch method). We used 10 initial points, a further 110 iterations for the sequential methods and 11 batches of size 10 for the batch methods. In our numerical experiments, we set $r^{\max} = 110\% r^{\min}$ (i.e., $r^{\max}$ is 10% higher than the minimal expected return $r^{\min}$). The results are given in Table 10.2.

	Sequential BO methods			**Batch BO methods**	
	CW-EI	**ACW-EI**	**2S-ACW-EI**	**KB-ACW-EI**	**2S-KB-ACW-EI**
1a CVaR (SD)	0.202 (0.013)	0.199 (0.013)	**0.184** (0.012)	0.199 (0.012)	0.191 (0.012)
1a Ex return (SD)	0.473 (0.012)	0.485 (0.012)	0.473 (0.012)	0.479 (0.012)	0.478 (0.012)
1b CVaR (SD)	0.266 (0.012)	0.253 (0.012)	**0.247** (0.012)	0.263 (0.014)	0.249 (0.013)
1b Ex return (SD)	0.581 (0.012)	0.577 (0.012)	0.561 (0.012)	0.580 (0.012)	0.567 (0.012)
2a CVaR (SD)	0.317 (0.013)	0.291 (0.015)	**0.275** (0.014)	0.302 (0.013)	0.287 (0.013)
2a Ex return (SD)	4.335 (0.013)	4.320 (0.012)	4.302 (0.013)	4.341 (0.012)	4.322 (0.012)
2b CVaR (SD)	0.336 (0.014)	0.320 (0.014)	**0.303** (0.013)	0.322 (0.013)	0.308 (0.013)
2b Ex return (SD)	4.427 (0.013)	4.428 (0.012)	4.417 (0.013)	4.433 (0.013)	4.420 (0.012)
3a CVaR (SD)	−0.094 (0.012)	−0.122 (0.014)	**−0.132** (0.013)	−0.102 (0.012)	−0.131 (0.014)
3a Ex return (SD)	2.105 (0.013)	2.030 (0.013)	1.938 (0.012)	2.082 (0.013)	1.970 (0.013)
3b CVaR (SD)	−0.075 (0.013)	−0.083 (0.013)	**−0.094** (0.014)	−0.064 (0.012)	−0.075 (0.013)
3b Ex return (SD)	2.113 (0.013)	2.075 (0.013)	2.056 (0.012)	2.125 (0.012)	2.089 (0.013)

Table 10.2: Optimal solution for the portfolio allocation problems: in each case, the best result among the methods is shown in bold. The standard deviations are given in parentheses.

Our proposed sequential methods outperform the standard BO approach for all

three examples, finding a lower CVaR objective value whilst meeting the feasibility condition. In addition, the two-stage approach produces better results than the one-stage approach. The same is true for the batch methods, where the two-stage method outperforms the one-stage method. The batch methods obtain better results than the standard sequential BO method but perform worse than the best sequential implementations. This is expected because the GP is only updated after a full batch has been identified, in contrast to the GP being updated after each new sample is proposed—as in the sequential approach.

To illustrate the results, we plot the current best solution's objective value after each iteration in Figure 10.2. Consistently, the best solution of 2S-ACW-EI decreases faster than the other two sequential methods. The two-stage batch method performs better than the standard implementation in all test cases.

Sensitivtiy to $r^{\max}$

A key parameter in the proposed algorithm is $r^{\max}$. Our numerical experiments found that setting $r^{\max} = 110\% r^{\min}$ generally works well. To test more rigorously how sensitive our proposed BO algorithm is to this parameter, we provide further numerical results obtained with $r^{\max} = 105\% r^{\min}$, in Table 10.3 and Figure 10.3. These results show that our proposed BO algorithms are not highly sensitive to the choice of $r^{\max}$.

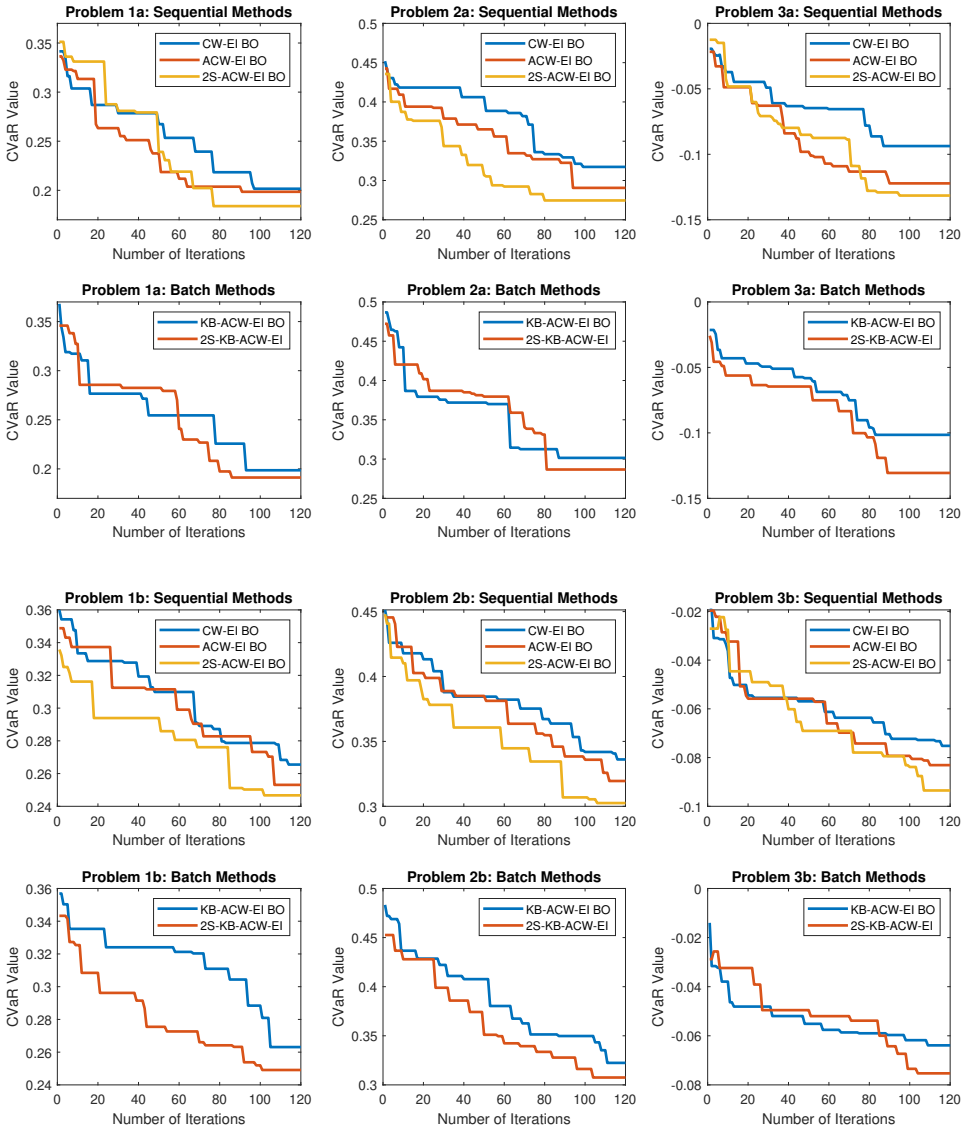


Figure 10.2: The best objective value obtained after each iteration for the portfolio allocation problems across the existing method (CW-EI BO) and the four new proposed methods.

	Sequential BO methods			Batch BO methods	
	CW-EI	**ACW-EI**	**2S-ACW-EI**	**KB-ACW-EI**	**2S-KB-ACW-EI**
1a CVaR (SD)	0.202 (0.013)	0.198 (0.011)	**0.188** (0.014)	0.201 (0.0.014)	0.194 (0.011)
1a Ex return (SD)	1.473 (0.012)	1.473 (0.018)	1.471 (0.013)	1.477 (0.016)	1.474 (0.017)
2a CVaR (SD)	0.317 (0.013)	0.299 (0.014)	**0.281** (0.014)	0.308 (0.011)	0.293 (0.017)
2a Ex return (SD)	5.335 (0.013)	5.324 (0.013)	5.317 (0.016)	5.331 (0.012)	5.323 (0.013)
3a CVaR (SD)	−0.094 (0.012)	−0.115 (0.016)	**−0.125** (0.013)	−0.112 (0.014)	−0.128 (0.015)
3a Ex return (SD)	3.105 (0.013)	3.103 (0.014)	3.083 (0.018)	3.102 (0.018)	3.061 (0.013)

Table 10.3: Optimal solutions for the portfolio allocation problems, with active region size of 5%

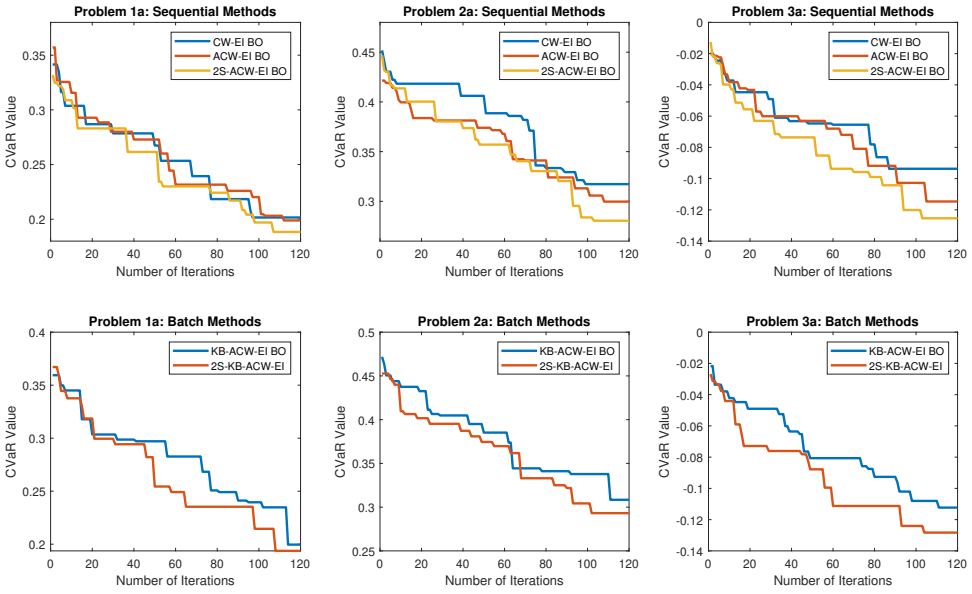


Figure 10.3: The best objective value obtained after each iteration with an active region size of 5%, for the base methodology (CW-EI BO) and our four proposed algorithms.

10.6.3 Advanced portfolio allocation example

To conclude this chapter, we provide the numerical results of the complete thesis framework applied to our core data set; seeking to find an optimal allocation to nine assets. The '2S-ACW-EI' BO algorithm is applied on the first available date of each month from January 2015 to December 2020—to propose an optimal allocation. For each month, MSMCS is applied to Bayesian ARMA-GARCH forecasts to obtain predicted CVaR and expected returns—which feed the BO algorithm. The portfolio is reallocated each month.

Based on an initial investment of \$1000 on January 2, 2015, we compare the performance of our complete framework (called *'Optimal Portfolio'*) with various r^{min} values against several other portfolio optimisation approaches:

- **Equal (Set Allocation) Portfolio:** This approach evenly distributes the initial investment across each asset on January 2, 2015, without any subsequent adjustments.

- **Equal (Monthly Rebalance) Portfolio:** In this strategy, the investment is initially divided equally among the assets on January 2, 2015, and then the portfolio is rebalanced at the start of each month to an equal weighting across all assets.

- **Markowitz (Set Allocation) Portfolio:** Based on Markowitz's portfolio theory, this method determines the optimal asset weightings on January 2, 2015, to maximise returns for a given risk level, without any subsequent adjustments.

- **Markowitz (Monthly Rebalance) Portfolio:** This strategy applies Markowitz's portfolio theory monthly, recalculating and adjusting asset weightings on the

first available date of each month to relocate and optimise the risk-return profile.

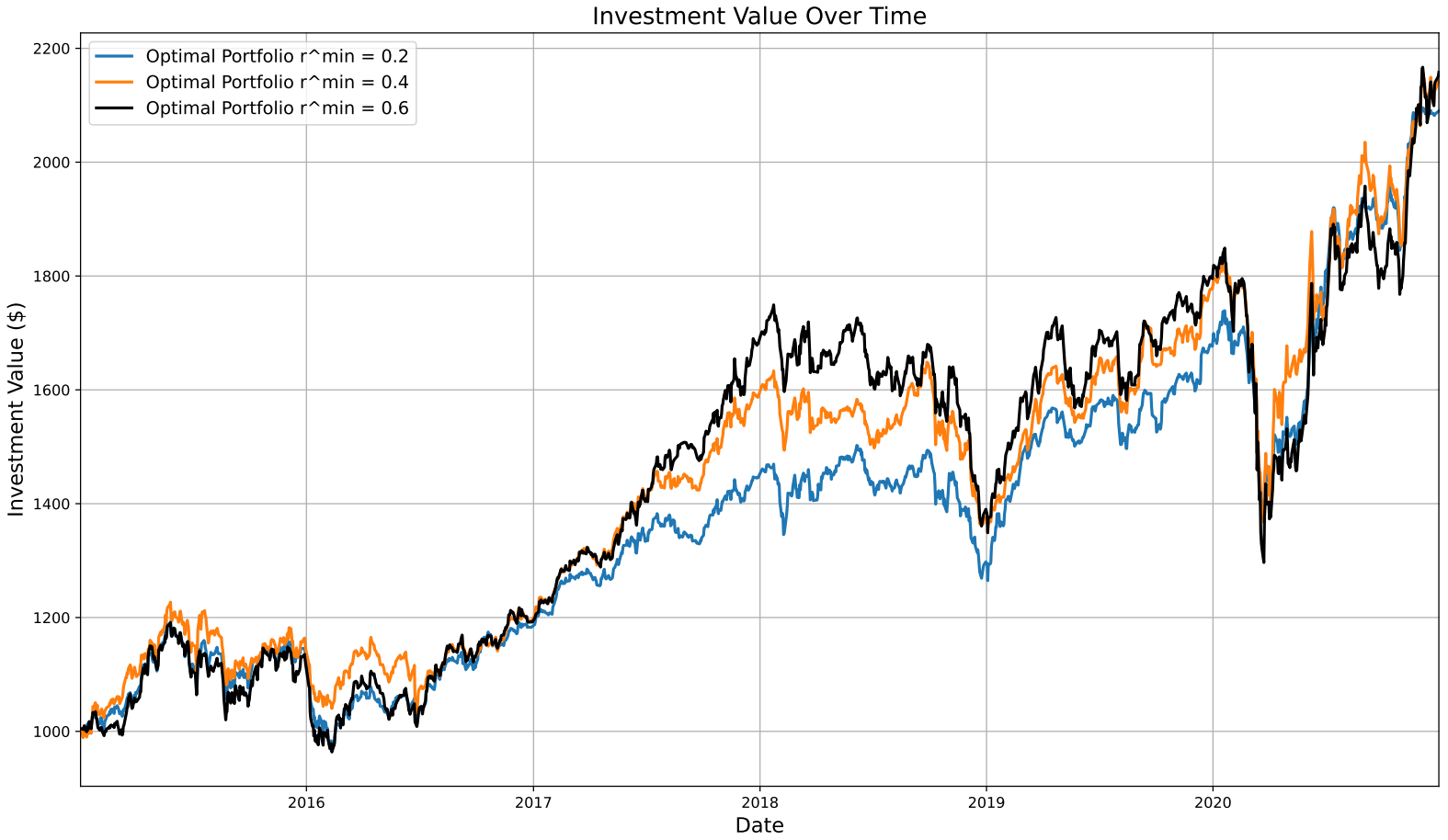
- **Minimum Variance Portfolio:** This strategy focuses on minimising risk by optimising asset weightings for the lowest possible portfolio variance, with the initial allocation set on January 2, 2015, without any subsequent adjustments.

- **Risk Parity Portfolio:** This portfolio strategy allocates investments based on risk, aiming to equalise the risk contribution of each asset to the portfolio, with the initial weights set on January 2, 2015, without any subsequent adjustments.

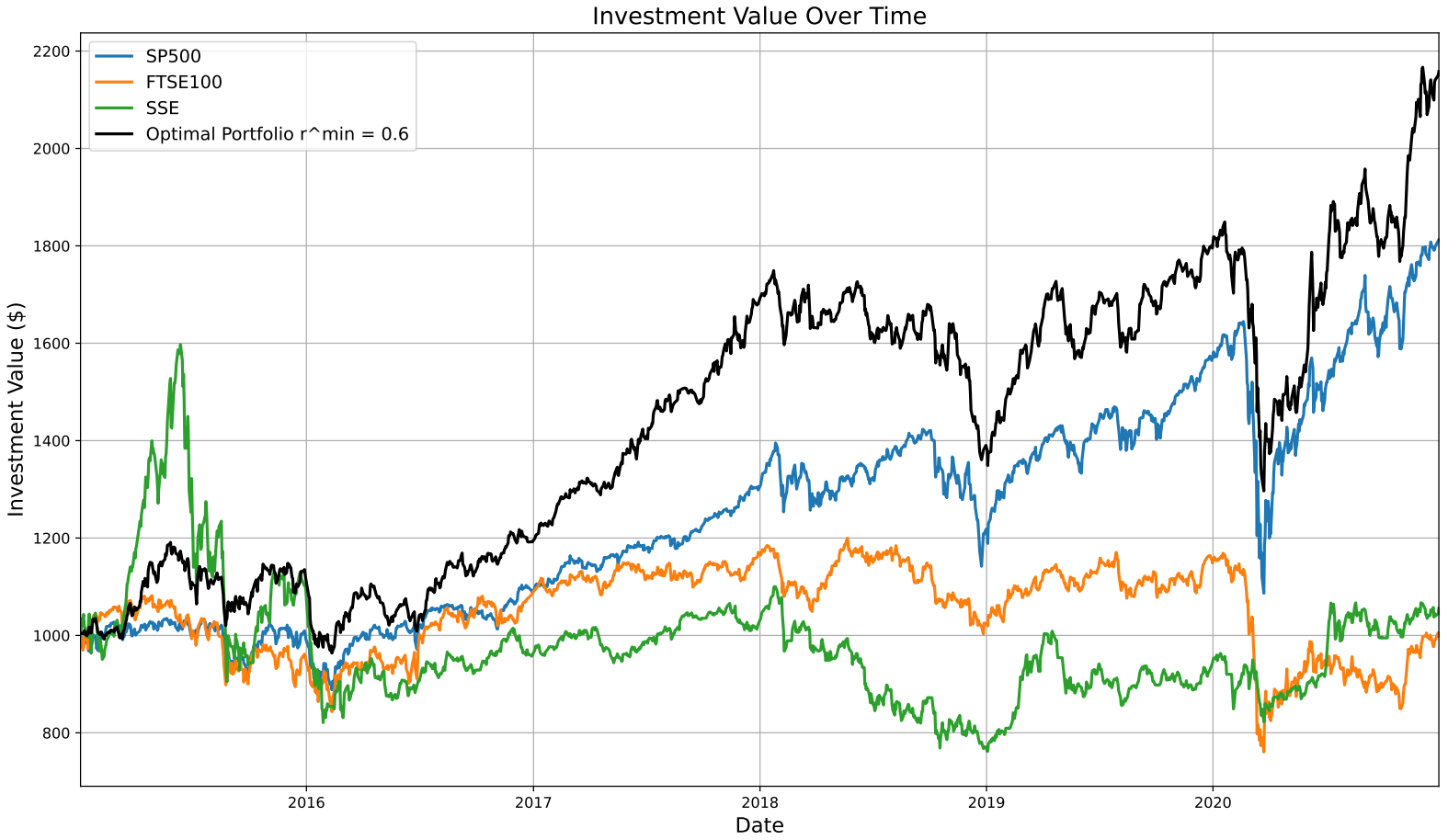
- **Maximum Diversification Portfolio:** This approach seeks to maximise diversification benefits by optimising the ratio of portfolio diversification to total risk, with initial weightings determined on January 2, 2015, without any subsequent adjustments.

The results are shown in the following pages, where our complete framework significantly outperforms all other portfolio optimisation methods in the comparative analysis. This performance can be attributed to the framework's robust consideration of both uncertainty and tail risk in financial markets, elements often overlooked or inadequately addressed by traditional models. By integrating Bayesian ARMA-GARCH models with the '2S-ACW-EI' Bayesian Optimisation algorithm, our approach dynamically adapts to changing market conditions and uncertainties, offering a more responsive asset allocation strategy. The framework's success in outperforming traditional and commonly used strategies underlines the achievement of the primary aim of this research: to find an optimal investment strategy for investors to allocate their resources in financial markets fraught with uncertainty.

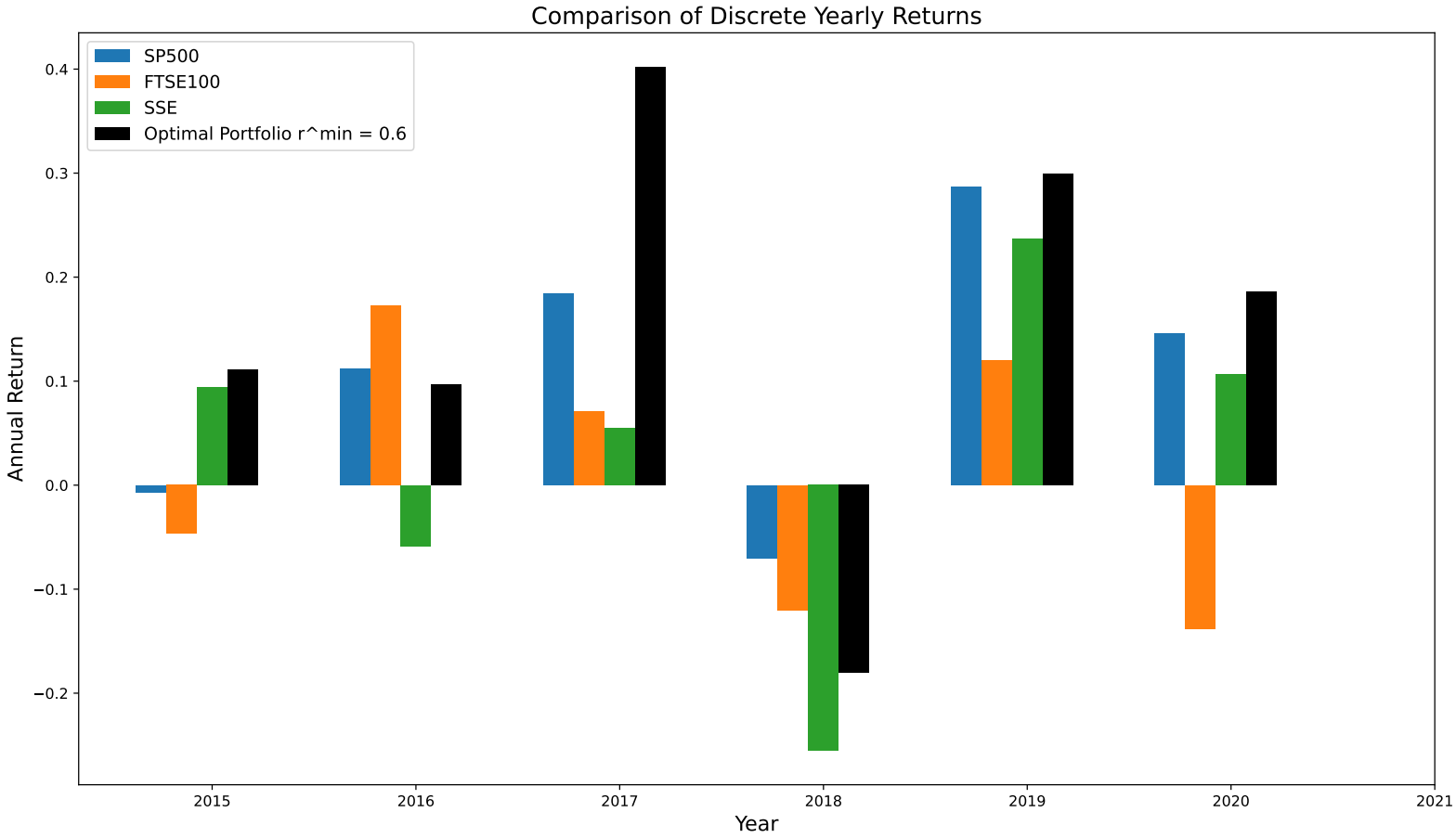
Optimal Portfolio Comparison



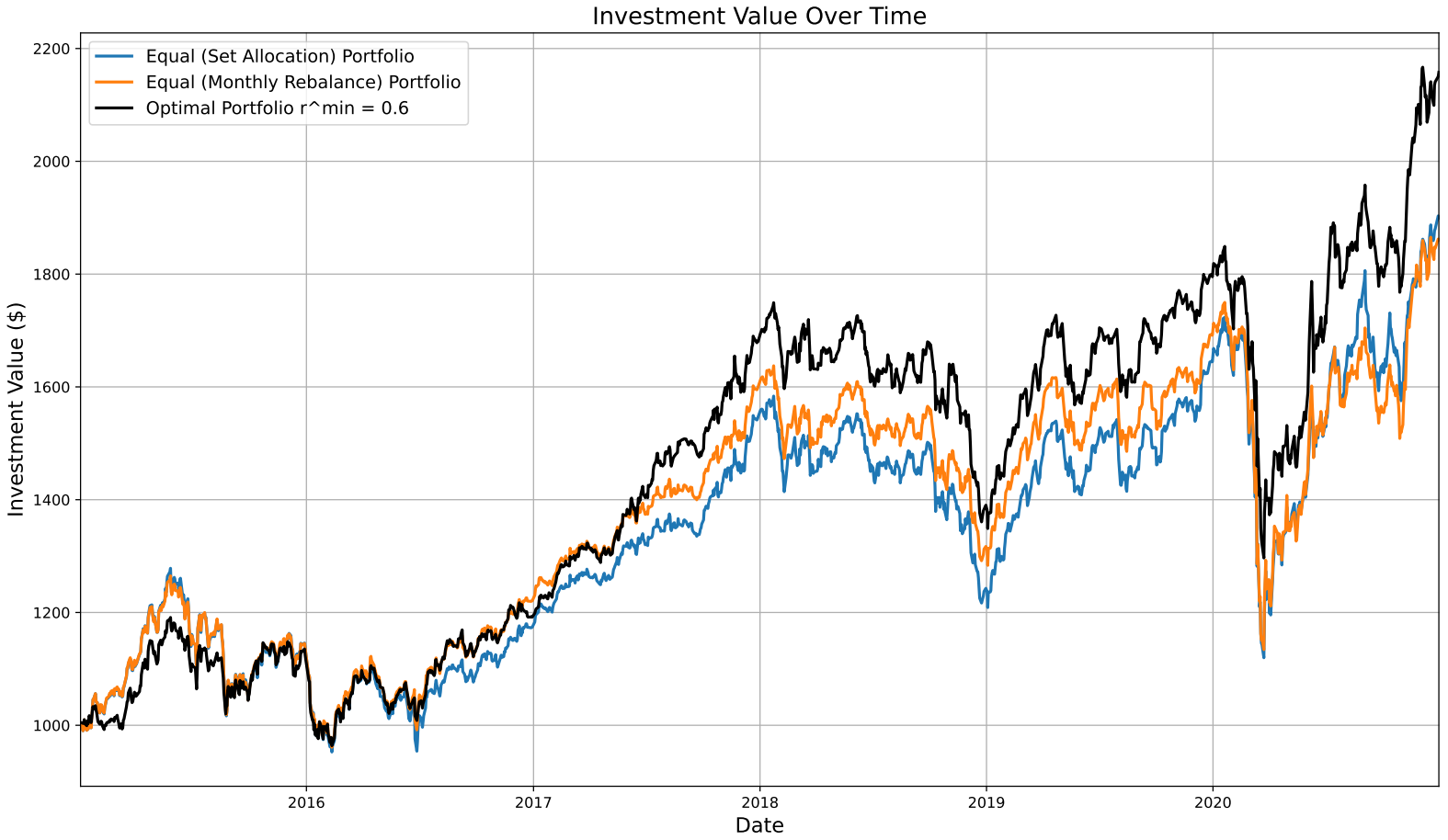
Major Indices vs. Optimal Portfolio



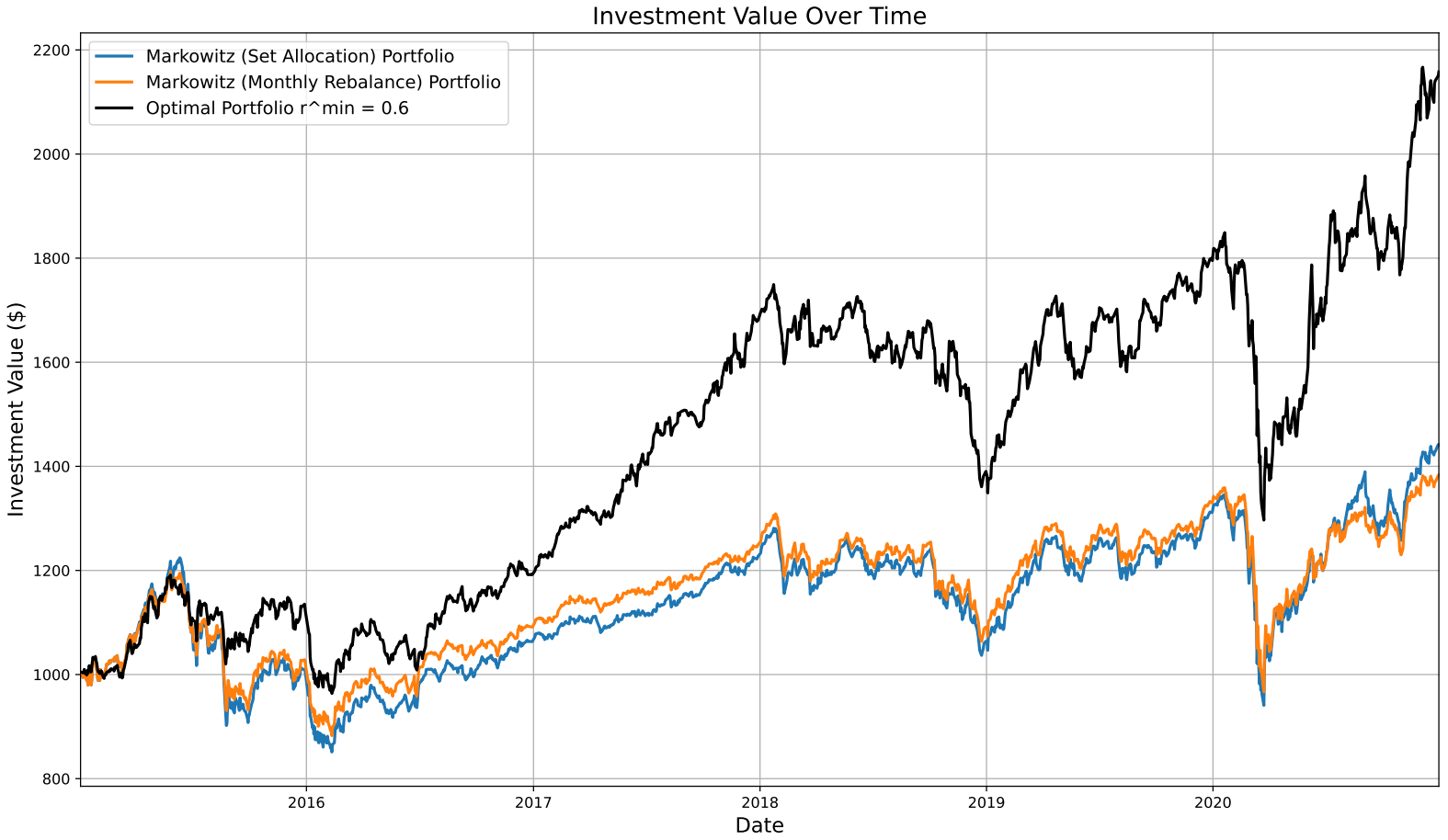
Major Indices vs. Optimal Portfolio



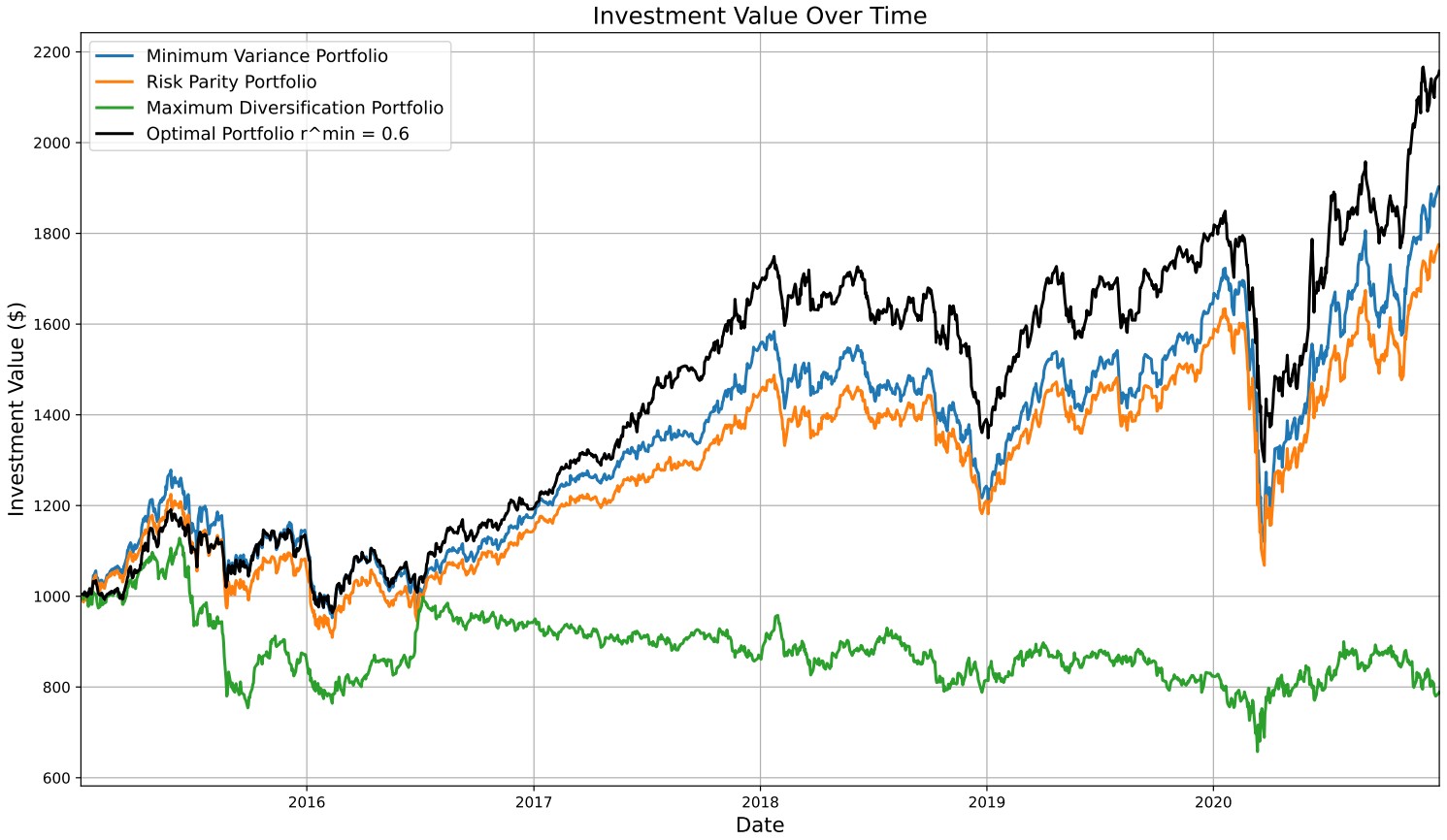
Equal Allocations vs. Optimal Portfolio



Markowitz vs. Optimal Portfolio



Advanced Methods vs. Optimal Portfolio



10.7 BO conclusion

This thesis has explored the twin challenges of forecasting asset returns and volatility in uncertain financial markets and measuring tail risks—building to our final thesis objective, to develop portfolio allocation algorithms which better reflect the complex dynamics of modern financial markets, removing unrealistic assumptions.

This chapter presented new Bayesian Optimisation algorithms that utilise Gaussian Processes' power to model the complex, non-linear relationships inherent in financial data. The chapter explored how BO algorithms could find an optimal portfolio to minimise CVaR under a minimum expected return requirement while accounting for financial market uncertainty through an environmental variable $\mathbf{Z}$.

We proposed several novel adaptations to standard Bayesian Optimisation procedures, specifically tailored to the nuances of portfolio allocation problems and designed to reduce the number of expensive objective function evaluations. The proposed BO algorithms included a new acquisition function, a two-stage point selection process, and a batch implementation to use parallel computing.

Our proposed BO algorithms consistently outperformed standard approaches in the numerical experiments, achieving lower CVaR while satisfying the minimum expected return constraint. These results underscore the effectiveness of our adaptations, particularly the two-stage optimisation process and the active constraint-weighted acquisition function, in efficiently navigating the complex landscape of portfolio allocation. We expect the proposed algorithms to be helpful in both financial portfolio allocation and similar problems arising from other fields, e.g., reinforcement learning (Ghosh et al., 2022).

Chapter 11

Conclusions

Financial markets are often assumed to be highly efficient, where prices reflect all available information, and the collective actions of investors ensure accurate pricing and effective risk management. These assumptions portray investors as completely rational, driven by the objective of maximising returns within their risk tolerance.

However, this conventional view faces substantial criticism, including in the works of Kay and King (2020), which critiques the standard economic and financial models for excessively relying on notions of rationality, efficiency, and the predictability of financial markets. Additionally, they emphasise the crucial distinction between risk and true uncertainty. As defined by Keynes (1936), risk is quantifiable and measurable through probability, whereas true uncertainty comprises unknown outcomes that resist probabilistic prediction, a factor paramount in financial decision-making.

Historical analysis conducted by Bond and Dow (2021) reveals a tendency among investors to concentrate on frequent yet relatively minor events. This focus often leads to the negligence of rare but significant tail events, which have significant financial implications. Recognising and quantifying tail risk is essential for enhanced preparedness against market disruptions and more effective resource allocation. Such an approach also deepens understanding of potential risks in financial markets.

Addressing the critical challenges of managing and modelling tail risk and uncertainty in financial markets is the cornerstone of this thesis. Traditional financial models frequently fall short in accounting for these factors, resulting in an underestimation of risks and the formulation of misguided investment strategies. This thesis set out to refine asset price forecasting, advance risk management methodologies, and optimise investment strategies to address the pivotal question: "How should investors allocate their resources in financial markets fraught with uncertainty?"

11.1 Asset returns forecasting

The first objective was to develop models that provide more accurate and robust asset price and volatility forecasts, accounting for uncertainty.

Chapter 2 established fundamental characteristics of financial time series, like stationarity—which ensures the stability of statistical properties through time—and autocorrelation, indicating that past returns may help predict future returns.

Chapter 3 introduced the ARMA-GARCH model for forecasting time series returns to take advantage of these properties; combining ARMA's proficiency in predicting the mean of a series with GARCH's ability to model and forecast changing volatility, offering a more robust framework for capturing linear relationships and volatility clustering in time series data. Additionally, we introduced TGARCH to enhance GARCH forecasting for asymmetric volatility modelling.

In Chapter 4, we achieved our first objective by developing a Bayesian approach to ARMA-(T)GARCH models, which allowed for the incorporation of prior knowledge and uncertainty measurements into financial forecasting models.

Bayesian ARMA-GARCH models present an avenue for further exploration, particularly in the domain of posterior sampling. The current methodology establishes posterior distributions of model parameters during training but relies on point

estimates–either real or forecasted mean returns-—to derive predictive posteriors of forecasted means and variances. This approach fails to fully exploit the Bayesian framework by neglecting the posterior distributions of the forecasted means. To address this limitation, it may be possible to employ Generalized Langevin Equations (GLE), stochastic differential equations designed to model systems with memory and history-dependent dissipation. By treating the problem as a hierarchical model where the posterior distributions of previous forecasted values inform subsequent values, we can better capture dependencies across different levels of the hierarchy. Utilising Langevin dynamics, which are stochastic processes for sampling from complex distributions, will enhance posterior sampling by incorporating memory effects and improving the modelling of dependencies within the hierarchy.

Bayesian methods, despite their robustness with finite outcomes, exhibit brittleness in continuous contexts with limited data (Owhadi et al., 2015). Posterior distributions can be highly sensitive to small changes in the model or prior distributions. Accurate Bayesian evidence estimation for model order selection is crucial to address this issue. When multiple models seem promising, averaging over a set of plausible models can enhance robustness. Hierarchical priors, which assign distributions to hyperparameters, can stabilise inferences, though at increased computational cost. Regularising likelihood functions can prevent overfitting, particularly in noisy data scenarios, thus enhancing the reliability of the models (Owhadi et al., 2015).

One limitation of this thesis is the choice to model each time series individually, establishing ARMA-GARCH models for each asset in isolation. This approach could fail to adequately account for the correlation structure which exists between some assets. The copula model explained in Ch. 5, allows for the decoupling of general risks and idiosyncratic risk, thereby allowing for better modelling of this correlation structure. Future research could take a more general vector-valued viewpoint, es-

tablishing a time series model which considers a collection of time series together - in the form of a vector - allowing for greater consideration of this correlation structure.

11.2 Risk management

The second objective was to develop new methodologies which better predict the probability and impact of tail events.

Chapter 5 introduced a quantitative framework for financial risk, specifically equity and credit risk. Concluding that CVaR was an essential measure of market risk and Student's t copula models were best for credit risk. Crucially, we found that equity risk, credit risk, and failure probability estimation necessitate the reconstruction of probability distributions to capture tail risk.

Chapters 6 & 7 analysed several Monte Carlo, importance sampling and more advanced methodologies which provide a variance-reduced estimator for a specific quantity associated with the distribution of our performance variable y, such as the probability of a given random event or failure probability. We sought to develop a new algorithm that obtains the entire probability distribution of $g(\mathbf{x})$, or $f(\mathbf{w}, \mathbf{Z})$—from which to calculate the expectation and other tail-based risk measures.

In Chapter 8, we introduced the novel multicanonical sequential Monte Carlo sampler (MSMCS) algorithm, which incorporates the multicanonical Monte Carlo method with the sequential Monte Carlo sampler—along with several other adaptations. Through numerical examples, we demonstrated that the proposed MSMCS method can outperform MC and standard MMC-MCMC. By combining MSMCS with the Bayesian ARMA-GARCH framework, we showed the power of this algorithm for both credit and equity risk management—achieving our second objective.

In the context of the Multicanonical Sequential Monte Carlo Sampler (SMCS), the selection of bin width poses a significant challenge. Accurate Conditional Value-

at-Risk (CVaR) values require narrow bins, which are computationally expensive, while wider bins, although easier to explore, may lead to issues with theta value updates. An adaptive bin width method could balance accuracy and computational efficiency, though this method must reset at each stage to maintain consistency in the theta update procedure–a topic which should be researched further.

Choosing optimal forward and backward kernels in SMCS is inherently challenging, often resulting in suboptimal choices that degrade efficiency. Adaptive kernel schemes that adjust parameters based on sampler performance can enhance efficiency. Optimisation algorithms can also be employed to fine-tune these kernel parameters, thereby improving the overall performance of the sampler. This presents a future avenue for research.

11.3 Optimal portfolio allocation

The third objective was to develop portfolio allocation algorithms which better reflect the complex dynamics of modern financial markets, removing unrealistic assumptions.

Chapter 9 introduced our overarching optimisation problem—utilising CVaR's properties—which set an objective of minimising CVaR under a minimum expected return constraint. The chapter explored various approaches to such risk-measure-based optimisation problems but found several limitations that we sought to address.

In Chapter 10, we presented four new BO algorithms specifically designed for our portfolio optimisation problem, accounting for uncertainty through an environmental random variable $\mathbf{Z}$. The proposed algorithms significantly reduce the number of evaluations of the expensive objective function by taking advantage of the special properties of portfolio optimisation problems. We developed a new acquisition function, a two-stage point selection process, and a batch implementation to take

advantage of parallel computing. Through simple numerical examples, we found that the '2S-ACW-EI' BO algorithm outperforms both existing methods and our other proposed algorithms—achieving our third objective.

Bayesian optimisation, particularly in the context of high-dimensional datasets, presents significant challenges. Large portfolio sizes result in large kernel matrices for Gaussian Processes (GP), leading to high computational and memory costs. Applying Principal Component Analysis (PCA) to the matrix can compress data, improving computational efficiency without compromising accuracy. Developing methods to achieve this without significant data loss is crucial for optimizing the acquisition function–as explored by Schafer et al. (2021).

The final numerical example in Chapter 10, draws on each aspect of our thesis to provide the numerical results for our core data set, using Bayesian ARMA-TGARCH models to forecast asset returns, MSMCS to reconstruct the probability distributions and assess tail risk, and BO to find optimal portfolio allocations. Our complete framework outperforms existing approaches and achieves the overall thesis objective of finding an optimal investment allocation considering both tail risk and uncertainty.

The proposed research directions aim to refine and enhance current methodologies for financial forecasting and risk management. By addressing the limitations of Bayesian ARMA-GARCH models, improving the robustness of the Multicanonical Sequential Monte Carlo Sampler, and optimising Bayesian approaches for high-dimensional datasets, future research can develop more accurate and efficient tools for navigating financial markets. These advancements will contribute to a deeper understanding of market dynamics, better management of tail risks, and more effective portfolio optimisation strategies.

List of References

Azam Abdollahi, Mehdi Azhdary Moghaddam, Seyed Arman Hashemi Monfared, Mohsen Rashki, and Yong Li. A refined subset simulation for the reliability analysis using the subset control variate. *Structural Safety*, 87:102002, 2020.

Amir Ahmadi-Javid and Malihe Fallah-Tafti. Portfolio optimization with entropic value-at-risk. *European Journal of Operational Research*, 279(1):225–241, 2019.

Chunrong Ai and Xiaohong Chen. Efficient estimation of models with conditional moment restrictions containing unknown functions. *Econometrica*, 71(6):1795–1843, 2003.

Mazin AM Al Janabi, Román Ferrer, and Syed Jawad Hussain Shahzad. Liquidity-adjusted value-at-risk optimization of a multi-asset portfolio using a vine copula approach. *Physica A: Statistical Mechanics and its Applications*, 536:122579, 2019.

E. C. Albagli, C. Hellwig, and A. Tsyvinski. A theory of asset pricing based on heterogeneous information. Yale University working paper, 2015.

Carol Alexander. *Market Models: A Guide to Financial Data Analysis*. John Wiley & Sons, 2008.

Timotheos Angelidis and Alexandros Benos. Liquidity adjusted value-at-risk based on the components of the bid-ask spread. *Applied Financial Economics*, 16(11): 835–851, 2006.

Jeffrey Annis, Nathan J Evans, Brent J Miller, and Thomas J Palmeri. Thermodynamic integration and steppingstone sampling methods for estimating Bayes factors: A tutorial. *Journal of mathematical psychology*, 89:67–86, 2019.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.

M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.

Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic engineering mechanics*, 16(4):263–277, 2001.

Siu Kui Au, J Ching, and JL Beck. Application of subset simulation methods to reliability benchmark problems. *Structural safety*, 29(3):183–193, 2007.

Turan G Bali and Nusret Cakici. Value at risk and expected stock returns. *Financial Analysts Journal*, 60(2):57–73, 2004.

Ray Ball. The theory of stock market efficiency: accomplishments and limitations. *Journal of Financial Education*, pages 1–13, 1996.

Thomas Bass. Recipe for disaster: The formula that killed wall street. *Wired*, Feb 2009. URL `www.wired.com/2009/02/wp-quant/`.

Achal Bassamboo, Sandeep Juneja, and Assaf Zeevi. Portfolio credit risk with extremal dependence: Asymptotic analysis and efficient simulation. *Operations Research*, 56(3):593–606, 2008.

Julien Bect, Ling Li, and Emmanuel Vazquez. Bayesian subset simulation. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):762–786, 2017.

Bernd A Berg. Introduction to multicanonical Monte Carlo simulations. *Fields Inst. Commun*, 26(1):1–24, 2000.

Bernd A Berg and Thomas Neuhaus. Multicanonical algorithms for first order phase transitions. *Physics Letters B*, 267(2):249–253, 1991.

Bernd A Berg and Thomas Neuhaus. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*, 68(1):9, 1992.

Alexandros Beskos, Dan Crisan, and Ajay Jasra. On the stability of sequential Monte Carlo methods in high dimensions. *The Annals of Applied Probability*, 24 (4):1396–1445, 2014.

Gino Biondini, William L Kath, and Curtis R Menyuk. Importance sampling for polarization-mode dispersion. *IEEE Photonics Technology Letters*, 14(3):310–312, 2002.

Zvi Bodie, Alex Kane, and Alan J Marcus. Investments. international edition. *New York, Boston, London*, 2002.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

Joshua J Bon, Anthony Lee, and Christopher Drovandi. Accelerating sequential Monte Carlo with surrogate likelihoods. *Statistics and Computing*, 31(5):1–26, 2021.

Philip Bond and James Dow. Failing to forecast rare events. *Journal of Financial Economics*, 142(3):1001–1016, 2021.

Alberto Bononi, Leslie A Rusch, Amirhossein Ghazisaeidi, Francesco Vacondio, and Nicola Rossi. A fresh look at multicanonical Monte Carlo from a telecom perspective. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–8. IEEE, 2009.

Karl Breitung. Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, 110(3):357–366, 1984.

Karl Breitung. The geometry of limit state function graphs and subset simulation: Counterexamples. *Reliability Engineering & System Safety*, 182:98–106, 2019.

B. Breon-Drish. On existence and uniqueness of equilibrium in a class of noisy rational expectations models. *Review of Economic Studies*, 2015. forthcoming.

Trevor S Breusch and Adrian R Pagan. A simple test for heteroskedasticity and random coefficient variation. *Econometrica: Journal of the Econometric Society*, pages 1287–1294, 1979.

E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Sait Cakmak, Raul Astudillo Marban, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. *Advances in Neural Information Processing Systems*, 33:20130–20141, 2020.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4): 907–929, 2004.

Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5): 899–924, 2007.

G. Chabakauri, K. Zachariadis, and K. Yuan. Multi-asset noisy rational expectations equilibrium with contingent claims. LSE working paper, 2021.

Gary Chamberlain. Asymptotic efficiency in estimation with conditional moment restrictions. *Journal of Econometrics*, 34(3):305–334, 1987.

Joshua CC Chan and Dirk P Kroese. Efficient estimation of large portfolio loss probabilities in t-copula models. *European Journal of Operational Research*, 205 (2):361–367, 2010.

Xinjuan Chen and Jinglai Li. A subset multicanonical Monte Carlo method for simulating rare failure events. *Journal of Computational Physics*, 344:23–35, 2017.

Zengjing Chen and Larry Epstein. Ambiguity, risk, and asset returns in continuous time. *Econometrica*, 70(4):1403–1443, 2002.

Kai Cheng, Zhenzhou Lu, Sinan Xiao, and Jingyu Lei. Estimation of small failure probability using generalized subset simulation. *Mechanical Systems and Signal Processing*, 163:108114, 2022.

Yin-Wong Cheung and Kon S Lai. Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280, 1995.

CNBC. 80% of the stock market is now on autopilot. CNBC, 2019. URL `https://www.cnbc.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.html`. Accessed: [Insert the date you accessed the article here].

T. Cogley and T. Sargent. The conquest of US inflation: Learning and robustness to model uncertainty. *Review of Economic Dynamics*, 8:528–563, 2005.

Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

Gregory Curtis. Modern portfolio theory and behavioural finance. *The Journal of Wealth Management*, 7(2):16–22, 2004.

Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective Bayesian optimization under input noise. In *International Conference on Machine Learning*, pages 4831–4866. PMLR, 2022a.

Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective Bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2022b.

Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436, 2006.

Anand Deo and Karthyek Murthy. Efficient black-box importance sampling for VaR and CVaR estimation. In *2021 Winter Simulation Conference (WSC)*, pages 1–12. IEEE, 2021.

Ove Ditlevsen and Henrik O. Madsen. Structural reliability methods. *Wiley*, 1996.

Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

Xiaoping Du and Wei Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design*, 126(2): 225–233, 2004.

Vincent Dubourg, Bruno Sudret, and Jean-Marc Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

Paul Embrechts, Filip Lindskog, and Alexander McNeil. Modelling dependence with copulas. *Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich*, 14, 2001.

Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling Extremal Events: for Insurance and Finance*. Springer Science & Business Media, 2013.

Paul Embrechts, Alexander Schied, and Ruodu Wang. Robustness in the optimization of risk measures. *Operations Research*, 70(1):95–110, 2022.

Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

David Eriksson and Matthias Poloczek. Scalable constrained Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. PMLR, 2021.

Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

Federal Reserve Bank of St. Louis. Russia's invasion of ukraine and its impact on stock prices. *Federal Reserve Bank of St. Louis*, 2022. URL `https://www.stlouisfed.org/on-the-economy/2022/may/russias-invasion-ukraine-impact-stock-prices`. Accessed: 2023-12-04.

Matthias Fischer, Thorsten Moser, and Marius Pfeuffer. A discussion on recent risk measures with application to credit risk: Calculating risk contributions and identifying risk concentrations. *Risks*, 6(4):142, 2018.

Isabella Fornacon-Wood, Hitesh Mistry, Corinne Johnson-Hart, Corinne Faivre-Finn, James PB O'Connor, and Gareth J Price. Understanding the differences between bayesian and frequentist statistics. *International journal of radiation oncology, biology, physics*, 112(5):1076–1082, 2022.

Rüdiger Frey and Alexander J McNeil. Modelling dependent defaults. Technical report, ETH Zurich, 2001.

Lukas Fröhlich, Edgar Klenske, Julia Vinogradska, Christian Daniel, and Melanie Zeilinger. Noisy-input entropy search for efficient robust Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2262–2272. PMLR, 2020.

Alexei A Gaivoronski and Georg Pflug. Value-at-risk in portfolio optimization: properties and computational approach. *Journal of risk*, 7(2):1–31, 2005.

G. Gao and Z. Song. Tail risk concerns everywhere. *Management Science*, 65: 3111–3130, 2018.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

John Geweke. Bayesian econometrics. *Economic policy and statistical analysis*, pages 11–15, 2001.

Laurent El Ghaoui, Maksim Oks, and Francois Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations research*, 51(4):543–556, 2003.

Supriyo Ghosh, Laura Wynter, Shiau Hong Lim, and Duc Thien Nguyen. Neural-progressive hedging: Enforcing constraints in reinforcement learning with stochastic programming. In *Uncertainty in Artificial Intelligence*, pages 707–717. PMLR, 2022.

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

Manfred Gilli and Evis Këllezi. A global optimization heuristic for portfolio choice with VaR and expected shortfall. In *Computational methods in decision-making, economics and finance*, pages 167–183. Springer, 2002.

Manfred Gilli, Evis Këllezi, and Hilda Hysi. A data-driven optimization heuristic for downside risk minimization. *Swiss Finance Institute Research Paper*, 2006.

Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

Paul Glasserman and Jingyi Li. Importance sampling for portfolio credit risk. *Management science*, 51(11):1643–1656, 2005.

Robert B. Gramacy and Herbert K. H. Lee. 229Optimization Under Unknown Constraints. In *Bayesian Statistics 9*. Oxford University Press, 10 2011. ISBN 9780199694587. doi: 10.1093/acprof:oso/9780199694587.003.0008. URL `https://doi.org/10.1093/acprof:oso/9780199694587.003.0008`.

Robert B Gramacy, Genetha A Gray, Sébastien Le Digabel, Herbert KH Lee, Pritam Ranjan, Garth Wells, and Stefan M Wild. Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11, 2016.

Peter L Green, LJ Devlin, RE Moore, RJ Jackson, J Li, and S Maskell. Increasing the efficiency of sequential Monte Carlo samplers through the use of approximately optimal l-kernels. *Mechanical Systems and Signal Processing*, 162:108028, 2022.

Quentin F Gronau, Andrew Heathcote, and Dora Matzke. Computing Bayes factors for evidence-accumulation models using Warp-III bridge sampling. *Behavior research methods*, 52(2):918–937, 2020.

Lars Peter Hansen. Nobel lecture: Uncertainty outside and inside economic models. *Journal of Political Economy*, 122(5):945–987, 2014.

Lars Peter Hansen and Tomas J Sargent. Robust control and model uncertainty. *The American Economic Review*, 91(2):60–66, 2001.

Lars Peter Hansen and Tomas J Sargent. *Recursive Robust Estimation and Control Without Commitment*. Journal of Economic Theory, 2007.

W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

George A Hazelrigg. A framework for decision-based engineering design. *Journal of mechanical design*, 120(4):653–658, 1998.

Matthias Heinkenschloss, Boris Kramer, Timur Takhtaganov, and Karen Willcox. Conditional-value-at-risk estimation via reduced-order models. *SIAM/ASA Journal on Uncertainty Quantification*, 6(4):1395–1423, 2018.

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential Monte Carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Ronald Hochreiter. An evolutionary computation approach to scenario-based risk-return portfolio optimization for general risk measures. In *Workshops on Applications of Evolutionary Computation*, pages 199–207. Springer, 2007.

Martin Hohenbichler and Rüdiger Rackwitz. New light on first- and second-order reliability methods. *Structural Safety*, 4(4):267–284, 1987.

Glyn Holton. Riskmetrics value-at-risk: Theory and practice, Jul 2016. URL `https://www.value-at-risk.net/riskmetrics/`.

Ronald Holzlööhner and Curtis R Menyuk. Use of multicanonical Monte Carlo simulations to obtain accurate bit error rates in optical communications systems. *Optics letters*, 28(20):1894–1896, 2003.

L Jeff Hong, Zhaolin Hu, and Guangwu Liu. Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 24(4):1–37, 2014.

John Hull. *Risk management and financial institutions,+ Web Site*, volume 733. John Wiley & Sons, 2012.

Yukito Iba, Nen Saito, and Akimasa Kitajima. Multicanonical MCMC for sampling rare events: an illustrative review. *Annals of the Institute of Statistical Mathematics*, 66(3):611–645, 2014.

International Monetary Fund IMF. Global financial stability report. *World Economic and Financial Surveys*, 2009.

Javed Iqbal, Sara Azher, and Ayesha Ijaz. Predictive ability of value-at-risk methods: evidence from the karachi stock exchange-100 index. Technical report, EERI Research Paper Series, 2010.

Philippe Jorion. *Value at Risk: The New Benchmark for Managing Financial Risk.* McGraw-Hill, 2000.

Herman Kahn and Andy W Marshall. Methods of reducing sample size in Monte Carlo computations. *Journal of the Operations Research Society of America*, 1 (5):263–278, 1953.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

Can B Kalayci, Okkes Ertenlice, and Mehmet Anil Akbay. A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, 125:345–368, 2019.

John Kay and Mervyn King. *Radical Uncertainty: Decision-Making for an Unknowable Future*. The Bridge Street Press, London, 2020.

Bryan Kelly and Hao Jiang. Tail risk and asset prices. *The Review of Financial Studies*, 27(10):2841–2871, 2014.

John Maynard Keynes. A treatise on probability. 1921.

John Maynard Keynes. *The General Theory of Employment, Interest, and Money*. Macmillan Cambridge University Press, for Royal Economic Society, London, 1936.

Frank H Knight. *Risk, Uncertainty, and Profit*. Houghton Mifflin, 1921.

Kevin H Knuth, Michael Habeck, Nabin K Malakar, Asim M Mubeen, and Ben Placek. Bayesian evidence and model selection. *Digital Signal Processing*, 47: 50–67, 2015.

Phaedon-Stelios Koutsourelakis, Helmuth J Pradlwarter, and Gerhart Iwo Schueller. Reliability of structures in high dimensions, part I: algorithms and applications. *Probabilistic Engineering Mechanics*, 19(4):409–417, 2004.

Slawomir Koziel, David Echeverría Ciaurri, and Leifur Leifsson. Surrogate-based methods. *Computational optimization, methods and algorithms*, pages 33–59, 2011.

Shunya Kusakawa, Shion Takeno, Yu Inatsu, Kentaro Kutsukake, Shogo Iwazaki, Takashi Nakano, Toru Ujihara, Masayuki Karasuyama, and Ichiro Takeuchi. Bayesian optimization for cascade-type multistage processes. *Neural Computation*, 34(12):2408–2431, 2022.

Remi Lam and Karen Willcox. Lookahead Bayesian optimization with inequality constraints. *Advances in Neural Information Processing Systems*, 30, 2017.

Nicolas Lartillot and Hervé Philippe. Computing Bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.

Jeong Eun Lee, Ross McVinish, and Kerrie Mengersen. Population Monte Carlo algorithm in high dimensions. *Methodology and Computing in Applied Probability*, 13(2):369–389, 2011.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. 2019.

David X Li. On default correlation: A copula function approach. *The Journal of Fixed Income*, 9(4):43–54, 2000.

Jing Li and Dongbin Xiu. Evaluation of failure probability via surrogate models. *Journal of Computational Physics*, 229(23):8966–8980, 2010.

Jing Li, Jinglai Li, and Dongbin Xiu. An efficient surrogate-based method for computing rare failure probability. *Journal of Computational Physics*, 230(24): 8683–8697, 2011.

Aurenice O Lima, Ivan T Lima, and Curtis R Menyuk. Error estimation in multicanonical Monte Carlo simulations with applications to polarization-mode-dispersion emulators. *Journal of lightwave technology*, 23(11):3781–3789, 2005.

Greta M Ljung and George EP Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

Fernando Llorente, Luca Martino, David Delgado, and Javier Lopez-Santiago. Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *SIAM Review*, 65(1):3–58, 2023.

Mejari Mallikarjuna and R Prabhakara Rao. Evaluation of forecasting methods from selected stock market returns. *Financial Innovation*, 5(1):1–16, 2019.

Harry M Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

Helmut Mausser and Dan Rosen. Beyond VaR: from measuring risk to managing risk. In *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr)(IEEE Cat. No. 99TH8408)*, pages 163–178. IEEE, 1999.

Ralph McKay and T Erle Keefer. Var is a dangerous technique. *Corporate Finance Searching for Systems Integration Supplement*, 9:30, 1996.

Warwick J McKibbin and Andrew Stoeckel. The global financial crisis: Causes and consequences. *Asian Economic Papers*, 9(1):54–86, 2010.

Alexander J McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, 2015.

Allan D R McQuarrie and Chih-Ling Tsai. *Regression and time series model selection*. World Scientific, 1999.

Xiao-Li Meng and Wing Hung Wong. Simulating ratios of normalizing constants via a simple identity: A theoretical exploration. *Statistica Sinica*, pages 831–860, 1996.

Nicholas Metropolis and Stanislaw Ulam. The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Caleb Miller, Jem N Corcoran, and Michael D Schneider. Rare events via cross-entropy population Monte Carlo. *IEEE Signal Processing Letters*, 2021.

Jonas Močkus. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

JP Morgan et al. Creditmetrics-technical document. *JP Morgan, New York*, 1997.

Saralees Nadarajah, Bo Zhang, and Stephen Chan. Estimation methods for expected shortfall. *Quantitative Finance*, 14(2):271–291, 2014.

Christian Naesseth, Fredrik Lindsten, and Thomas Schon. Nested sequential Monte Carlo methods. In *International Conference on Machine Learning*, pages 1292–1301. PMLR, 2015.

Radford M Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 139–188. Chapman and Hall/CRC, 2011.

Whitney K Newey. Efficient instrumental variables estimation of nonlinear models. *Econometrica*, 58(4):809–837, 1990.

Duc Manh Nguyen, Hoai An Le Thi, and Tao Pham Dinh. A cross-entropy method for value-at-risk constrained optimization. In *Asian Conference on Intelligent Information and Database Systems*, pages 442–451. Springer, 2011.

Quoc Phong Nguyen, Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Optimizing conditional value-at-risk of black-box functions. *Advances in Neural Information Processing Systems*, 34, 2021a.

Quoc Phong Nguyen, Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Value-at-risk optimization with Gaussian processes. In *International Conference on Machine Learning*, pages 8063–8072. PMLR, 2021b.

Anna Orlik and Laura Veldkamp. Understanding uncertainty shocks and the role of black swans. Technical report, National bureau of economic research, 2014.

Houman Owhadi, Clint Scovel, and Tim Sullivan. On the brittleness of Bayesian inference. *siam REVIEW*, 57(4):566–582, 2015.

Iason Papaioannou, Wolfgang Betz, Kilian Zwirglmaier, and Daniel Straub. MCMC algorithms for subset simulation. *Probabilistic Engineering Mechanics*, 41:89–103, 2015.

Andrew J Patton. A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4–18, 2012.

M Hashem Pesaran and Allan Timmermann. Market timing and return prediction under model instability. *Journal of Empirical Finance*, 8(4):495–510, 2001.

Georg Ch Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. *Probabilistic constrained optimization: Methodology and applications*, pages 272–281, 2000.

Victor Picheny, Henry Moss, Léonard Torossian, and Nicolas Durrande. Bayesian quantile and expectile optimisation. In *Uncertainty in Artificial Intelligence*, pages 1623–1633. PMLR, 2022.

Traian A Pirvu. Portfolio optimization under the value-at-risk constraint. *Quantitative Finance*, 7(2):125–136, 2007.

HJ Pradlwarter, GI Schueller, Phaedon-Stelios Koutsourelakis, and Dimos C Charmpis. Application of line sampling simulation method to reliability benchmark problems. *Structural Safety*, 29(3):208–221, 2007.

Svetlozar T Rachev. *Handbook of heavy tailed distributions in finance*. Elsevier, 2003.

Rüdiger Rackwitz. Reliability analysis—a review and some perspectives. *Structural Safety*, 23(4):365–395, 2001.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

Gareth O Roberts and Jeffrey S Rosenthal. General state space markov chains and MCMC algorithms. 2004.

Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic processes and their applications*, 49(2):207–216, 1994.

R Tyrrell Rockafellar and Stan Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Samir Saissi Hassani and Georges Dionne. The new international regulation of market risk: Roles of VaR and CVaR in model validation. 2021.

Florian Schafer, Timothy John Sullivan, and Houman Owhadi. Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *Multiscale Modeling & Simulation*, 19(2):688–730, 2021.

GI Schuëller, HJ Pradlwarter, and Phaedon-Stelios Koutsourelakis. A critical appraisal of reliability estimation procedures for high dimensions. *Probabilistic engineering mechanics*, 19(4):463–474, 2004.

John Skilling. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Didier Sornette, Guilherme Demos, Qun Zhang, Peter Cauwels, Vladimir Filimonov, and Qunzhi Zhang. Real-time prediction and post-mortem analysis of the shanghai 2015 stock market bubble and crash. *Swiss finance institute research paper*, (15-31), 2015.

LF South, AN Pettitt, and CC Drovandi. Sequential Monte Carlo samplers with independent Markov chain monte carlo proposals. *Bayesian Analysis*, 14(3):753–776, 2019.

L. Straub and R. Ulbricht. Endogenous uncertainty and credit crunches. Harvard University working paper, 2021.

Lihua Sun and L Jeff Hong. A general framework of importance sampling for value-at-risk and conditional value-at-risk. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 415–422. IEEE, 2009.

Robert H Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607, 1986.

Armin Tabandeh, Gaofeng Jia, and Paolo Gardoni. A review and assessment of importance sampling methods for reliability analysis. *Structural Safety*, 97:102216, 2022.

Nassim Taleb. The black swan: Why don't we learn that we don't learn. *NY: Random House*, 1145, 2005.

Qihe Tang, Zhaofeng Tang, and Yang Yang. Sharp asymptotics for large portfolio losses under extreme risks. *European Journal of Operational Research*, 276(2): 710–722, 2019.

Ruey S Tsay. *Analysis of financial time series*. John wiley & sons, 2005.

R. Turner, D. Eriksson, M. Mccourt, and I. Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning. *arXiv:2104.10201v2 [cs.LG]*, Aug 2020. 31 Aug 2021.

Douglas N VanDerwerken and Scott C Schmidler. Parallel Markov chain monte carlo. *arXiv preprint arXiv:1312.7479*, 2013.

Ziqi Wang, Marco Broccardo, and Junho Song. Hamiltonian Monte Carlo methods for subset simulation in reliability analysis. *Structural Safety*, 76:51–67, 2019.

Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer Science & Business Media, 2006.

Jo Yung Wong. *Theory of ground vehicles*. John Wiley & Sons, 2008.

David Wozabal. Value-at-risk optimization using the difference of convex algorithm. *OR spectrum*, 34(4):861–883, 2012.

Jiangqi Wu, Linjie Wen, Peter L Green, Jinglai Li, and Simon Maskell. Ensemble Kalman filter based Sequential Monte Carlo Sampler for sequential Bayesian inference. *arXiv preprint arXiv:2012.08848*, 2020.

Keyi Wu and Jinglai Li. A surrogate accelerated multicanonical Monte Carlo method for uncertainty quantification. *Journal of Computational Physics*, 321:1098–1109, 2016.

Y-T Wu, HR Millwater, and TA Cruse. Advanced probabilistic structural analysis method for implicit performance functions. *AIAA journal*, 28(9):1663–1669, 1990.

Panos Xidonas, Ralph Steuer, and Christis Hassapis. Robust portfolio optimization: a categorized bibliographic review. *Annals of Operations Research*, 292(1):533–552, 2020.

Wei Xie, Paul O Lewis, Yujiang Fan, Lynn Kuo, and Ming-Hui Chen. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic Biology*, 60(2):150–160, 2010.

Yue Xing, Tony Sit, and Hoi Ying Wong. Variance reduction for risk measures with importance sampling in nested simulation. *Quantitative Finance*, 22(4):657–673, 2022. doi: 10.1080/14697688.2021.1985730.

David Yevick. Multicanonical communication system modeling-application to PMD statistics. *IEEE Photonics Technology Letters*, 14(11):1512–1514, 2002.

Assaf Zeevi and Roy Mashal. Beyond correlation: Extreme co-movements between financial assets. *Available at SSRN 317122*, 2002.

Konstantin Zuev. Subset simulation method for rare event estimation: an introduction. *arXiv preprint arXiv:1505.03506*, 2015.

Konstantin M Zuev, James L Beck, Siu-Kui Au, and Lambros S Katafygiotis. Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers & structures*, 92:283–296, 2012.

Steve Zymler, Daniel Kuhn, and Berç Rustem. Worst-case value at risk of nonlinear portfolios. *Management Science*, 59(1):172–188, 2013.

Appendix A
Financial Analysis Details

A.1 Normality test results

Asset	Statistic (Test)	Value	p-value	Normality
SP500	Shapiro-Wilk	0.86	1.68e-44	Reject H0
	Anderson-Darling	74.76	-	Reject H0
	Jarque-Bera	32078.90	0.0	Reject H0
	Kolmogorov-Smirnov	0.48	0.0	Reject H0
AAPL	Shapiro-Wilk	0.93	2.04e-34	Reject H0
	Anderson-Darling	36.64	-	Reject H0
	Jarque-Bera	4999.44	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
ALK	Shapiro-Wilk	0.88	8.13e-42	Reject H0
	Anderson-Darling	48.21	-	Reject H0
	Jarque-Bera	30792.30	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
FTSE100	Shapiro-Wilk	0.93	9.74e-35	Reject H0
	Anderson-Darling	33.64	-	Reject H0
	Jarque-Bera	11652.40	0.0	Reject H0
	Kolmogorov-Smirnov	0.48	0.0	Reject H0
SHEL	Shapiro-Wilk	0.87	1.18e-42	Reject H0
	Anderson-Darling	50.58	-	Reject H0
	Jarque-Bera	45470.18	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
TW	Shapiro-Wilk	0.91	8.06e-38	Reject H0
	Anderson-Darling	32.84	-	Reject H0
	Jarque-Bera	45114.81	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
SSE	Shapiro-Wilk	0.90	1.21e-38	Reject H0
	Anderson-Darling	62.08	-	Reject H0
	Jarque-Bera	5458.80	0.0	Reject H0
	Kolmogorov-Smirnov	0.48	0.0	Reject H0
PING	Shapiro-Wilk	0.94	1.95e-31	Reject H0
	Anderson-Darling	33.31	-	Reject H0
	Jarque-Bera	3390.91	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
SHEN	Shapiro-Wilk	0.95	2.31e-30	Reject H0
	Anderson-Darling	38.65	-	Reject H0
	Jarque-Bera	1191.71	1.68e-259	Reject H0
	Kolmogorov-Smirnov	0.46	0.0	Reject H0

Table A.1: Normality Tests for Core Data Daily Log Returns

Asset	Statistic (Test)	Value	p-value	Normality
Portfolio	Shapiro-Wilk	0.88	2.47e-42	Reject H0
	Anderson-Darling	47.23	-	Reject H0
	Jarque-Bera	20448.32	0.0	Reject H0
	Kolmogorov-Smirnov	0.41	0.0	Reject H0
SP500	Shapiro-Wilk	0.86	8.83e-44	Reject H0
	Anderson-Darling	60.66	-	Reject H0
	Jarque-Bera	22875.53	0.0	Reject H0
	Kolmogorov-Smirnov	0.48	0.0	Reject H0
AAPL	Shapiro-Wilk	0.99	3.67e-13	Reject H0
	Anderson-Darling	5.04	-	Reject H0
	Jarque-Bera	118.69	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
ALK	Shapiro-Wilk	0.87	7.03e-43	Reject H0
	Anderson-Darling	39.72	-	Reject H0
	Jarque-Bera	26332.40	0.0	Reject H0
	Kolmogorov-Smirnov	0.47	0.0	Reject H0
FTSE100	Shapiro-Wilk	0.88	3.27e-42	Reject H0
	Anderson-Darling	36.23	-	Reject H0
	Jarque-Bera	22794.89	0.0	Reject H0
	Kolmogorov-Smirnov	0.45	0.0	Reject H0
SHEL	Shapiro-Wilk	0.89	3.56e-41	Reject H0
	Anderson-Darling	35.79	-	Reject H0
	Jarque-Bera	20982.15	0.0	Reject H0
	Kolmogorov-Smirnov	0.43	0.0	Reject H0
TW	Shapiro-Wilk	0.90	1.91e-38	Reject H0
	Anderson-Darling	36.22	-	Reject H0
	Jarque-Bera	8159.00	0.0	Reject H0
	Kolmogorov-Smirnov	0.39	0.0	Reject H0
SSE	Shapiro-Wilk	0.96	2.31e-27	Reject H0
	Anderson-Darling	24.28	-	Reject H0
	Jarque-Bera	1050.45	0.0	Reject H0
	Kolmogorov-Smirnov	0.42	0.0	Reject H0
PING	Shapiro-Wilk	0.99	4.67e-15	Reject H0
	Anderson-Darling	4.42	-	Reject H0
	Jarque-Bera	333.00	0.0	Reject H0
	Kolmogorov-Smirnov	0.41	0.0	Reject H0
SHEN	Shapiro-Wilk	0.98	1.02e-17	Reject H0
	Anderson-Darling	8.47	-	Reject H0
	Jarque-Bera	273.53	0.0	Reject H0
	Kolmogorov-Smirnov	0.39	0.0	Reject H0

Table A.2: Normality Tests for Core Data Monthly Log Returns

A.2 Augmented Dickey-Fuller (ADF) rest results

Asset	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
SP500	3.77e-29	1.09e-12	3.63e-29	3.34e-30	8.99e-29	1.36e-23	3.18e-30	3.33e-30	1.05e-11	6.81e-30	7.41e-04
AAPL	1.03e-27	5.56e-16	6.92e-07	8.14e-28	4.40e-17	4.18e-29	2.53e-27	2.14e-08	1.46e-27	6.36e-30	1.64e-05
ALK	8.43e-29	1.14e-29	8.29e-30	8.60e-30	2.17e-30	1.56e-08	6.70e-28	4.79e-27	1.91e-14	1.15e-29	3.56e-03
FTSE100	1.04e-27	2.22e-26	3.20e-30	4.88e-13	1.88e-11	7.52e-29	5.21e-15	7.59e-16	4.88e-30	6.83e-26	1.18e-06
SHEL	5.12e-28	6.01e-26	7.74e-30	1.90e-27	5.55e-29	5.19e-28	8.63e-22	1.27e-27	5.93e-30	5.75e-27	2.07e-26
TW	1.02e-27	1.05e-14	3.24e-18	2.76e-16	1.38e-22	2.93e-24	5.13e-10	6.48e-30	3.86e-30	6.14e-29	5.19e-15
SSE	9.27e-29	9.05e-30	4.90e-29	1.64e-06	1.61e-02	1.39e-03	6.04e-14	7.20e-29	1.06e-07	1.65e-04	1.20e-28
PING	3.28e-19	1.50e-08	2.32e-29	4.44e-27	1.09e-4	5.89e-27	3.79e-23	8.61e-14	1.06e-10	1.60e-05	1.27e-10
SHEN	6.25e-29	2.52e-26	2.94e-11	2.55e-13	2.25e-15	4.30e-04	0.0	1.24e-22	1.38e-29	1.44e-25	4.67e-27

Table A.3: ADF Test p-values for Core Data Daily Log Returns.

Asset	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
SP500	0.0677	0.0159	0.0442	3.70e-05	0.0069	0.0101	0.0601	0.0012	0.0427	0.0042	0.0044
AAPL	0.0064	0.0010	0.3154	0.0169	0.0136	0.0518	0.0002	0.0018	0.4606	0.0268	0.1011
ALK	0.0259	0.1421	0.0481	0.1669	0.0002	0.0155	0.2315	0.0001	0.0060	0.0044	0.0441
FTSE100	0.0284	0.0049	0.0208	0.0000	0.0002	0.0197	0.0819	0.0001	0.0552	0.0016	0.0085
SHEL	0.0595	0.0163	0.0208	0.0000	0.0015	0.0136	0.0002	0.0209	0.0242	0.0104	0.0179
TW	0.2028	0.0145	0.0670	0.0091	0.0087	0.0427	0.0095	0.0600	0.0202	0.0854	0.0047
SSE	0.0036	0.1049	0.3072	0.0018	0.5780	0.0935	0.0003	0.0488	0.0001	0.0278	0.0198
PING	0.0922	0.0082	0.0002	0.0917	0.2383	0.1175	0.1119	0.0093	0.0337	0.0003	0.0640
SHEN	0.0653	0.0015	0.0423	0.0003	0.1823	0.0412	0.0000	0.5396	0.0115	0.0270	0.2902

Table A.4: ADF Test p-values for Core Data Monthly Log Returns.

Appendix B
Bayesian Evidence Results

B.1 ARMA Bayesian evidence

		Model Order p															
		0	**1**	**2**	**3**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**13**	**14**	**15**	**16**	**22**
Model Order q	**0**	8577.10	8609.74	8618.59	8617.63	8618.94	8627.21	8638.93	8646.09	8651.13	8650.14	8653.30	8652.30	8652.82	8652.12	8651.51	8651.55
	1	8604.34	8614.59	8617.59	8616.67	8618.05	8624.83	8636.74	8644.25	8650.00	8649.03	8652.20	8651.20	8651.63	8651.00	8650.39	8651.55
	2	8617.85	8618.19	8615.82	8615.36	8616.79	8623.16	8634.11	8642.40	8648.70	8647.76	8650.83	8649.85	8650.47	8649.89	8649.20	8650.15
	3	8618.67	8622.00	8616.49	8615.36	8615.64	8622.15	8633.18	8626.47	8647.79	8646.85	8649.88	8648.89	8649.55	8648.98	8648.29	8648.51
	4	8618.13	8623.56	8612.13	8608.94	8646.01	8629.27	8633.11	8646.85	8645.79	8482.20	8649.07	8648.08	8648.73	8648.14	8647.56	8647.43
	5	8618.25	8625.29	8624.33	8642.88	8602.34	8635.74	8631.52	8639.18	8643.94	8644.98	8646.12	8647.11	8647.84	8647.20	8646.53	8645.67
	6	8619.94	8631.81	8643.62	8653.21	8629.83	8632.07	8633.16	8604.02	8643.94	8643.45	8646.81	8645.83	8646.79	8646.23	8645.56	8644.92
	7	8626.09	8638.30	8637.63	8646.54	8649.09	8646.55	8645.66	8649.94	8649.67	8648.53	8645.61	8644.59	8645.73	8645.28	8644.41	8643.24
	8	8633.56	8640.74	8639.80	8646.93	8650.63	8650.81	8650.89	8649.92	8643.33	8647.39	8644.24	8643.34	8644.83	8644.51	8644.08	8642.01
	9	8642.43	8642.23	8641.55	8645.59	8648.59	8650.87	8649.05	8648.55	8641.85	8637.77	8642.20	8641.84	8643.54	8643.54	8642.43	8639.93
	13	8646.67	8648.61	8649.34	8648.67	8647.82	8648.28	8652.89	8651.82	8597.06	8640.08	8638.09	8637.40	8644.66	8639.83	8638.57	8637.12
	15	8648.85	8650.86	8649.63	8649.03	8652.40	8651.19	8654.36	8653.28	8652.21	8652.92	8638.73	8633.34	8637.35	8637.72	8636.65	8634.13

Table B.1: S&P 500

		Model Order p									
		0	**1**	**2**	**3**	**5**	**8**	**9**	**14**	**18**	**22**
Model Order q	**0**	7253.81	7257.09	7256.17	7255.80	7254.21	7259.32	7263.88	7262.72	7260.75	7261.58
	1	7257.03	7256.27	7255.14	7254.78	7253.19	7258.01	7262.83	7261.69	7259.81	7260.61
	2	7256.11	7255.36	7254.44	7253.77	7252.20	7256.99	7261.82	7260.68	7258.82	7259.60
	3	7255.72	7254.34	7253.75	7252.75	7251.20	7255.95	7260.84	7259.67	7257.82	7258.59
	5	7254.14	7255.75	7258.02	7261.95	7255.40	7258.67	7260.19	7261.58	7255.83	7256.60
	8	7256.96	7260.78	7260.85	7260.12	7252.95	7250.23	7256.07	7255.83	7252.87	7253.80
	9	7261.86	7261.35	7259.55	7259.68	7257.39	7256.63	7257.60	7255.31	7251.94	7252.75
	14	7260.39	7259.37	7259.11	7257.71	7259.63	7258.04	7257.29	7254.01	7252.27	7247.83
	19	7262.02	7261.05	7260.62	7260.90	7258.54	7256.54	7256.90	7252.12	7244.60	7243.17

Table B.2: AAPL

		Model Order p													
		0	**1**	**2**	**3**	**5**	**6**	**7**	**8**	**9**	**11**	**12**	**13**	**15**	**17**
Model Order q	**0**	6407.63	6407.88	6410.38	6411.05	6409.69	6412.27	6415.00	6417.00	6418.73	6425.73	6427.50	6428.87	6428.63	6430.43
	1	6407.77	6396.01	6406.71	6410.06	6408.66	6411.19	6413.94	6415.91	6417.75	6424.82	6426.43	6427.86	6427.64	6429.44
	2	6410.12	6410.19	6411.31	6409.25	6407.57	6409.95	6412.62	6414.85	6416.58	6423.95	6425.60	6426.74	6426.53	6431.90
	3	6411.31	6409.14	6407.81	6418.71	6406.58	6408.78	6414.44	6406.49	6415.55	6422.98	6424.50	6425.74	6425.59	6427.52
	5	6409.46	6412.47	6407.47	6406.38	6409.99	6411.42	6411.83	6412.95	6414.53	6423.95	6422.87	6424.25	6427.04	6425.52
	6	6410.65	6411.31	6413.13	6411.21	6422.29	6409.46	6416.32	6414.01	6416.27	6422.74	6423.55	6423.08	6422.61	6426.61
	7	6412.76	6414.23	6414.89	6412.02	6418.55	6417.59	6414.63	6413.60	6414.18	6419.84	6422.66	6423.25	6427.05	6423.76
	8	6414.30	6412.99	6414.27	6411.87	6421.55	6422.11	6420.10	6418.70	6412.43	6427.68	6427.11	6423.82	6422.02	6422.75
	9	6415.94	6415.86	6416.94	6416.25	6422.84	6423.86	6421.42	6428.35	6424.28	6429.69	6428.60	6424.85	6422.39	6422.28
	11	6424.61	6424.52	6422.93	6422.92	6422.35	6428.49	6427.42	6425.84	6426.64	6427.72	6426.34	6429.11	6420.49	6420.66
	12	6425.56	6424.66	6423.79	6422.59	6421.92	6427.86	6426.56	6426.15	6426.55	6426.60	6426.49	6427.16	6421.18	6420.67
	13	6426.29	6425.18	6425.87	6425.07	6425.85	6427.36	6426.31	6425.91	6426.08	6426.52	6426.97	6425.35	6431.04	6419.92
	17	6431.27	6430.23	6427.64	6427.65	6429.86	6428.89	6427.60	6429.50	6428.53	6351.49	6421.34	6275.08	6420.35	6418.83

Table B.3: ALK

		Model Order p										
		0	**1**	**2**	**3**	**5**	**6**	**7**	**8**	**9**	**13**	**24**
Model Order q	**0**	8729.35	8728.48	8727.69	8726.73	8726.10	8728.22	8731.53	8738.87	8739.35	8739.32	8736.09
	1	8728.49	8728.99	8727.06	8725.73	8725.10	8727.26	8730.61	8737.93	8738.35	8738.34	8735.08
	2	8727.70	8727.02	8725.35	8724.35	8724.14	8726.35	8729.73	8737.00	8737.40	8737.31	8734.06
	3	8726.71	8726.00	8724.35	8723.35	8721.34	8725.41	8728.76	8730.30	8736.45	8736.30	8733.02
	5	8726.54	8733.19	8731.29	8733.73	8728.53	8729.88	8734.12	8734.37	8734.37	8734.01	8731.04
	6	8729.28	8734.83	8736.73	8738.29	8726.45	8722.81	8726.04	8732.93	8733.22	8733.14	8730.10
	7	8732.08	8736.45	8735.73	8734.14	8734.56	8742.00	8744.88	8731.73	8732.09	8732.25	8728.88
	8	8738.36	8737.98	8736.38	8735.27	8733.29	8732.47	8732.44	8730.89	8730.86	8731.38	8727.92
	13	8739.06	8738.63	8735.93	8736.35	8736.17	8734.90	8738.75	8696.17	8730.49	8726.82	8723.09
	24	8729.15	8733.15	8732.39	8734.64	8732.34	8732.07	8730.66	8729.85	8729.55	8726.48	8713.66

Table B.4: FTSE100

		Model Order p									
		0	**1**	**2**	**3**	**5**	**8**	**13**	**19**	**22**	**24**
Model Order q	**0**	7452.63	7458.45	7458.77	7457.78	7457.80	7471.29	7474.38	7474.10	7474.06	7473.66
	1	7458.86	7458.86	7457.77	7456.77	7456.85	7470.52	7473.37	7473.08	7472.93	7472.77
	2	7458.79	7457.80	7456.55	7455.78	7455.93	7469.56	7472.44	7472.12	7471.85	7471.67
	3	7457.78	7456.93	7455.77	7454.78	7454.93	7468.54	7471.46	7471.11	7470.85	7470.67
	5	7458.27	7460.13	7458.14	7458.23	7461.66	7466.94	7469.44	7469.28	7468.99	7471.38
	7	7460.32	7462.13	7460.32	7459.06	7466.38	7471.79	7467.35	7467.25	7466.86	7469.50
	8	7469.26	7468.17	7466.89	7466.44	7472.37	7468.34	7465.71	7465.89	7465.78	7465.43
	13	7472.79	7471.83	7471.02	7469.59	7466.67	7470.57	7467.84	7461.75	7461.07	7460.79
	19	7474.34	7473.30	7471.94	7471.80	7472.31	7470.72	7470.14	7463.99	7455.68	7454.87
	22	7473.45	7472.84	7472.52	7471.13	7471.50	7468.78	7465.94	7462.06	7451.87	7453.60
	23	7475.01	7474.14	7474.30	7473.15	7471.80	7469.74	7466.82	7460.86	-7459.39	7456.18

Table B.5: SHEL

		Model Order p								
		0	**1**	**2**	**3**	**4**	**5**	**8**	**16**	**23**
Model Order q	**0**	6432.56	6436.01	6435.14	6434.53	6436.05	6436.43	6434.96	6431.87	6431.95
	1	6436.08	6435.10	6434.13	6433.47	6434.94	6435.42	6433.96	6430.76	6430.76
	2	6435.11	6434.12	6434.52	6432.48	6433.21	6434.29	6432.95	6429.76	6429.74
	3	6434.36	6435.07	6432.40	6435.34	6433.28	6433.38	6431.96	6428.77	6428.73
	4	6435.50	6435.57	6436.06	6435.04	6433.47	6432.17	6431.01	6427.74	6432.14
	5	6436.25	6435.27	6435.25	6434.31	6432.44	6431.35	6430.00	6426.76	6430.57
	8	6434.96	6434.44	6432.90	6432.96	6431.96	6433.06	6434.17	6423.65	6425.84
	16	6430.65	6429.75	6428.72	6426.35	6425.38	6426.03	6428.70	6423.52	6421.42
	23	6432.67	6432.84	6434.88	6433.77	6433.32	6432.94	6429.27	6418.96	6419.10

Table B.6: TW

		Model Order p											
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**13**	**21**	**23**
Model Order q	**0**	8043.95	8044.29	8044.27	8043.47	8046.25	8045.25	8050.54	8052.27	8054.48	8061.30	8060.50	8061.34
	1	8044.36	8043.82	8042.89	8042.44	8045.25	8044.24	8049.67	8051.16	8053.41	8060.37	8059.54	8060.73
	2	8044.21	8043.18	8042.25	8041.36	8044.14	8043.14	8048.60	8050.12	8052.44	8059.37	8058.52	8059.75
	3	8043.21	8041.89	8036.28	8032.20	8048.11	8044.94	8048.36	8052.77	8053.33	8058.42	8057.56	8058.77
	4	8045.30	8044.27	8051.02	8051.95	8051.67	8052.51	8053.23	8048.04	8051.06	8057.42	8056.59	8057.71
	5	8044.25	8043.34	8052.48	8050.68	8050.60	8048.14	8048.56	8054.88	8050.89	8056.52	8055.61	8056.74
	6	8050.68	8049.88	8051.94	8050.82	8049.98	8053.07	8053.56	8054.70	8053.57	8055.54	8054.49	8055.44
	7	8053.52	8053.43	8052.40	8053.30	8054.31	8054.25	8053.61	8051.50	8055.84	8049.03	8053.57	8054.70
	8	8056.71	8055.23	8053.71	8052.47	8055.14	8055.48	8055.23	8055.54	8055.13	8053.84	8052.46	8053.75
	13	8061.26	8060.28	8059.08	8058.15	8057.66	8056.13	8061.10	8061.04	8056.67	8033.45	8049.56	8048.10
	15	8060.25	8059.25	8058.63	8057.94	8056.90	8057.34	8062.18	8063.04	8063.76	7999.17	8045.63	8046.05
	21	8058.59	8056.23	8059.13	8059.67	8059.47	8057.99	8059.61	8058.84	8061.56	8057.27	8044.45	8039.55
	23	8060.84	8058.76	8058.08	8057.92	8057.13	8055.34	8056.91	8055.83	8058.82	8056.16	8041.77	8046.31

Table B.7: SSE

		Model Order p							
		0	**1**	**2**	**3**	**4**	**5**	**8**	**23**
Model Order q	**0**	7098.04	7101.45	7100.45	7099.69	7100.30	7099.86	7099.89	7089.54
	1	7101.45	7100.45	7099.45	7098.73	7099.35	7098.89	7098.87	7088.52
	2	7100.44	7101.26	7098.44	7097.73	7096.73	7097.89	7097.88	7087.52
	3	7099.88	7098.83	7097.83	7096.79	7097.38	7096.93	7096.98	7086.55
	4	7100.68	7100.99	7100.00	7098.79	7097.62	7095.97	7095.77	7085.54
	5	7100.13	7099.83	7098.89	7097.77	7096.04	7095.07	7094.73	7084.52
	7	7100.22	7098.82	7098.05	7097.19	7097.99	7094.14	7097.15	7082.54
	8	7099.22	7098.05	7097.06	7096.48	7096.68	7095.60	7097.12	7081.58
	23	7089.55	7088.53	7087.49	7086.72	7085.69	7085.23	6952.33	7073.75

Table B.8: PING

		Model Order p												
		0	**1**	**2**	**3**	**5**	**7**	**8**	**11**	**12**	**13**	**20**	**21**	**23**
Model Order q	**0**	6380.00	6382.57	6389.32	6388.37	6391.76	6391.52	6390.95	6390.16	6393.13	6394.09	6391.22	6394.15	6393.44
	1	6383.15	6386.11	6388.42	6387.35	6390.78	6390.47	6393.93	6389.25	6391.96	6393.11	6390.25	6393.14	6392.41
	2	6389.51	6389.71	6382.14	6386.62	6389.57	6389.54	6388.96	6388.09	6390.90	6392.03	6389.10	6392.09	6391.46
	3	6388.92	6388.90	6387.36	6388.17	6392.74	6388.52	6387.94	6387.13	6389.90	6391.03	6388.08	6391.16	6390.45
	5	6390.69	6388.94	6393.84	6392.47	6386.29	6387.87	6386.09	6385.28	6389.61	6389.67	6386.37	6390.09	6389.49
	7	6391.60	6391.20	6391.66	6391.11	6389.47	6388.59	6388.31	6387.31	6386.29	6387.48	6386.16	6386.89	6386.31
	8	6391.53	6390.73	6390.74	6390.22	6390.06	6387.87	6383.30	6383.90	6386.17	6387.46	6384.59	6386.03	6385.30
	12	6392.06	6392.60	6391.52	6395.01	6391.76	6393.37	6384.28	6384.39	6384.36	6384.79	6381.15	6382.70	6381.31
	13	6393.79	6391.98	6390.45	6392.96	6390.87	6392.24	6391.75	6390.04	6389.41	6379.41	6385.51	6385.39	6381.39
	20	6390.17	6389.85	6389.59	6387.83	6386.33	6391.03	6388.97	6389.22	6386.69	6377.86	6374.84	6373.76	6380.15
	21	6391.18	6390.15	6389.79	6388.33	6387.24	6389.15	6388.79	6381.15	6380.78	6379.08	6376.64	6378.60	6382.33

Table B.9: SHEN