AALBORG UNIVERSITY
STUDENT REPORT

MASTER'S THESIS
MATHEMATICS-ECONOMICS

Neural Network Methods for Gas Storage Valuation

A Least Squares Monte Carlo Study with Application to Gas Storage Denmark

27 May 2026

Dept. of Mathematical Sciences
Thomas Manns Vej 23, 3rd floor
DK-9220 Aalborg Ø
http://math.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Neural Network Methods for Gas Storage Valuation

Theme:
A Least Squares Monte Carlo Study with Application to Gas Storage Denmark

Project Period:
01 Sep 2025 – 27 May 2026

Project Group:
M10-01

Authors:
Rasmus Flaathen Sønder

Supervisors:
Esben Høg

Numbered Pages: 100

Date of Completion:
27 May 2026

Abstract:

This thesis examines the valuation of natural gas storage contracts as a stochastic dynamic optimisation problem, comparing six continuation value approximation methods within the Least Squares Monte Carlo framework: polynomial OLS regression, spline regression, and nodewise and joint-input variants of the multi-layer perceptron (MLP) and the Kolmogorov–Arnold Network (KAN). Spot price dynamics are modelled using a one-factor mean-reverting model calibrated to Trading Hub Europe prices, and the framework is applied to the Danish storage system of Gas Storage Denmark (GSD) and a stylised Dutch salt cavern. Polynomial OLS and spline regression outperform both neural network approaches in the configurations studied. For the GSD case, OLS yields €123.481M while the best neural network, the joint-input MLP, reaches €107.101M, constituting a gap of approximately 13%. The near-affine inventory dependence of the continuation value favours polynomial regression and limits the benefit of larger networks. The study is limited by the one-factor price model, neural network training cost, and the computational inefficiency of the nodewise KAN, which is slower than OLS by a factor of roughly 600. Overall, the results indicate that classical LSMC with polynomial regression is both more accurate and orders of magnitude faster than neural network alternatives for the class of gas storage problems studied.

Preface

This thesis is written by Rasmus Flaathen Sønder in the period from 1 September 2025 to 27 May 2026. The primary focus of this report is the comparison of continuation value approximation methods, ranging from classical polynomial regression to neural networks, within the Least Squares Monte Carlo framework for natural gas storage valuation.

In this project, the coding and data processing is done in Python. The code and data are available on GitHub: `https://github.com/rsoender02/masters-thesis-gas-storage-valuation.git`.

Generative AI has been used to improve clarity and coherence and to assist with coding tasks.

References follow the APA referencing style and are formatted as

(Author, Year, Chapter or Page number) .

The author would like to thank the project supervisor, Esben Høg, for his guidance throughout the process. Further, the author extends his gratitude to Mikkel Mandrup for advice on the direction of this thesis.

Signature:

Rasmus Sønder

Rasmus Flaathen Sønder
rsond21@student.aau.dk

Nomenclature

$\mathbb{R}_0^+$	The set of non-negative real numbers
C^k	The set of k-times continuously differentiable functions
D^β	Multi-index notation for partial derivatives
$\|\cdot\|_{C^m}$	The C^m-norm measuring the magnitude of derivatives up to order m
$\mathbb{P}$	The physical probability measure
$\mathbb{Q}$	The risk-neutral probability measure
$\mathbb{E}_{\mathbb{P}}$	Expectation under the physical measure
$\mathbb{E}_{\mathbb{Q}}$	Expectation under the risk-neutral measure
$S(t)$	Spot price at time t
S_{50}	Median simulated spot price at a given time
$X(t)$	Log-spot price, $X(t) = \ln S(t)$
$\tilde{\mu}(t)$	Long-run log-price mean under the physical measure $\mathbb{P}$
$\tilde{\mu}_{\mathbb{Q}}(t)$	Risk-neutral long-run mean function under $\mathbb{Q}$
κ	Mean-reversion rate in the one-factor spot-price model
σ	Volatility parameter in the one-factor spot-price model
$F(0,t)$	Forward price for delivery at time t
$v(t)$	Storage inventory level at the start of day t
$v^{\min}(t)$	Minimum inventory constraint at time t
$v^{\max}(t)$	Maximum inventory constraint at time t
$\Delta v(t)$	Volume change (control action) on day t
δ	Continuously compounded discount rate
$\hat{V}_0$	Estimated storage value at the initial time
$C(t,v,S)$	Continuation value as a function of time, inventory, and spot price
V_{in}	In-sample value estimate
V_{out}	Out-of-sample value estimate
Q	Number of basis functions in the LSMC regression
α_c	Coarse-grid spacing in the inventory discretisation
α_f	Fine-grid spacing in the inventory discretisation
N_c	Number of coarse-grid inventory nodes
N_f	Number of fine-grid inventory nodes
r	Grid refinement factor linking the coarse and fine inventory grids
γ_r	Learning rate at gradient descent iteration r
L	Network depth (number of layers)
G	KAN spline grid size
k	Spline order in the KAN specification

Contents

1 | Introduction

Gas storage has traditionally served different operational purposes in the natural gas industry. Storage facilities allow production capacity to be shifted across time, helping match relatively stable supply with seasonal demand, particularly during winter peak consumption periods. They also provide a buffer against unexpected fluctuations in demand or supply, for example by supplementing pipeline capacity during periods of heavy use or compensating for disruptions caused by unusual weather, industrial demand swings, natural disasters, or geopolitical crises. Historically, in regulated market settings, storage was therefore primarily used by utilities as an operational balancing instrument to ensure system reliability.

Over the past decades, however, the liberalisation and deregulation of natural gas markets have fundamentally altered this role. Storage facilities are now increasingly accessible for commercial use in addition to operational balancing, which enables market participants to exploit predictable seasonal price variations and short-term trading opportunities. As storage capacity has been separated from traditional utility operations and made actively tradeable in spot and forward markets, it has evolved into an economic asset whose value must be quantified. The 2022 European gas supply crisis, in which benchmark prices reached unprecedented levels, prompted the EU to impose mandatory seasonal storage filling targets and reinforced the need for rigorous valuation of storage assets.

From a financial perspective, gas storage represents a portfolio of real options. At each decision time, the holder may inject gas, withdraw gas, or remain inactive, subject to physical constraints on capacity and flow rates. The value of storage therefore arises from the operational flexibility to respond optimally to evolving price conditions. Unlike a standard European derivative, the payoff is not determined at a single maturity but is generated through a sequence of decisions under uncertainty. The valuation problem is inherently dynamic and path-dependent.

Two fundamental characteristics of gas prices drive storage value. First, predictable components such as seasonality create systematic spreads between periods, enabling intrinsic strategies based on the forward curve. Second, stochastic fluctuations around these seasonal patterns generate additional optionality. A purely forward-based valuation captures only the value implied by currently observed spreads. A spot-based valuation, by contrast, incorporates the effect of volatility and mean reversion, and therefore reflects the full flexibility embedded in the asset. The magnitude of the extrinsic value depends critically on the assumed price dynamics.

Storage valuation can be formulated as a stochastic control problem with state variables given by the spot price and the current inventory level. The admissible control set is restricted by injection and withdrawal limits as well as minimum and maximum inventory constraints. The resulting dynamic program resembles that of American-style options, yet differs in essential ways. The decision space contains multiple actions at each time, the payoff may be both positive and negative, and the inventory introduces an additional state dimension. Analytical solutions are generally unavailable, which necessitates numerical approximation.

Monte Carlo methods provide a flexible framework for this task. In particular, regression-based approaches such as the Least Squares Monte Carlo method enable the approximation of continuation values in high-dimensional settings. An important aspect is the separation between the specification of the stochastic price model and the optimisation of the exercise strategy. This modular structure allows for realistic modelling of gas price dynamics, incorporating factors such as volatility, mean reversion, and seasonality, while simultaneously accommodating complex operational constraints. For storage assets with binding physical constraints, such flexibility is essential.

Accurate valuation is not only relevant for short-term trading decisions but also for long-term investment planning. Decisions regarding working volume, injection and withdrawal capacities, and technological design must be evaluated against their marginal impact on storage value. Since these characteristics interact nonlinearly with market dynamics, reliable quantitative methods are required to assess sensitivity with respect to both price parameters and operational constraints.

The continuation value function, which encodes the expected future payoff of the storage asset at each state, must be approximated over a domain shaped by stochastic price dynamics and physical constraints. Classical regression methods impose a fixed functional form on this approximation. Neural network methods, specifically multi-layer perceptrons (MLPs) and Kolmogorov–Arnold Networks (KANs), are nonparametric alternatives capable of representing nonlinear functions without specifying basis functions in advance. Whether this additional flexibility yields improved valuations in the gas storage setting, and at what computational cost, motivates the following research questions.

1. How do MLP- and KAN-based continuation value approximators perform relative to classical regression methods for gas storage valuation within the LSMC framework, in terms of valuation accuracy and computational cost?

2. How do the methods compare in empirical convergence behaviour and sensitivity to simulation design, and what does this imply for practical method selection?

1.1 Literature review

Breslin et al. (2008) review gas storage valuation methods, identifying three approaches distinguished by their treatment of price uncertainty: intrinsic, rolling intrinsic, and spot-based. Maragos (2002) observes that storage value increases with the amount of risk taken, with spot-based methods yielding the highest return under the highest risk exposure. The spot-based approach frames the storage problem as a stochastic dynamic optimisation task, structurally analogous to American option pricing but extended to continuous control, multiple admissible actions at each decision time, and an endogenous inventory state that evolves with the chosen strategy. The Monte Carlo framework of Boogert and De Jong (2007) is the foundational spot-based treatment of gas storage valuation and the direct methodological predecessor of the present work.

Accurate storage valuation requires realistic modelling of spot price dynamics. Mean-reverting processes have become standard in commodity markets. The one-factor model of Schwartz (1997) captures short-term mean reversion, while the two-factor model of Schwartz and Smith (2000) separates short-term deviations from long-term equilibrium dynamics. These models reflect empirical features such as seasonality, volatility clustering, and term structure dynamics. The interaction between mean reversion and storage value is subtle. Higher volatility increases extrinsic value, whereas stronger mean reversion can either improve trade timing or suppress the extreme price movements that generate arbitrage opportunities. Empirical evidence from Boogert and De Jong (2007) suggests that volatility is often the dominant driver of storage value. A limitation shared by one-factor mean-reverting models is the inability to accommodate sudden price spikes or regime shifts, both of which have been prominent features of European gas markets. This motivates the two-factor framework of Schwartz and Smith (2000) and, more generally, jump-diffusion specifications, though tractability concerns often favour the simpler one-factor representation. Calibration of the simulated price process to observed forward prices requires a smooth term structure extracted from discrete contract quotes. Benth et al. (2007) and Lim and Xiao (2002) develop spline-based methods for constructing smooth forward curves from average-based commodity contracts, providing the calibration target for the risk-neutral drift.

Longstaff and Schwartz (2001) introduced the least-squares Monte Carlo algorithm for American option pricing, approximating continuation values by regressing discounted future payoffs onto basis functions of the current state. Clément et al. (2002) established rigorous convergence results, showing that the estimator converges to the true option value as the number of simulated paths and the number of basis functions grow, provided the basis is dense in the relevant function space. Boogert and De Jong (2007) generalised the algorithm to the gas storage problem, noting that the relationship between volume, price, and continuation value is often insufficiently smooth for reliable regression over the joint state and recommending separate regressions for each discretised volume level. The convergence results of Clément et al. (2002) were established for a simpler option-pricing setting, and whether they extend directly to the storage problem, where the endogenous inventory state and multiple admissible actions alter the structure of the dynamic programme, is not established here. Boogert and De Jong (2007) provide empirical evidence that the algorithm nonetheless converges rapidly.

The theoretical foundation for neural network approximation is the Universal Approximation Theorem (Hornik et al., 1989). This result establishes that a single-hidden-layer network with a non-linear activation can approximate any Borel-measurable function to arbitrary accuracy, making networks a natural nonparametric alternative to fixed polynomial bases in the LSMC regression step. A recently proposed architecture, the Kolmogorov–Arnold Network (Liu et al., 2025), places learnable univariate B-spline functions on the edges of the network graph, in contrast to MLPs where fixed activation functions are applied at the nodes and edges carry only weighted sums. KANs are motivated by the Kolmogorov–Arnold representation theorem (Kolmogorov, 1957), which states that any continuous multivariate function admits a finite superposition of univariate continuous functions and addition. Liu et al. (2025) report improved predictive accuracy and parameter efficiency relative to multi-layer perceptrons on several benchmark tasks, and suggest that the architecture may offer advantages when the target function has smooth compositional structure.

Together, these contributions establish LSMC as the standard framework for spot-based gas storage valuation, with price dynamics and basis function choice as the primary modelling degrees of freedom. The existing literature demonstrates that gas storage valuation can be formulated as a stochastic dynamic programming problem and effectively approximated through simulation-based regression methods (Boogert and De Jong, 2007). Neural-network approaches have been proposed for high-dimensional stochastic control problems (Becker et al., 2020), but there is limited empirical evidence on whether such methods actually outperform classical regression-based LSMC in realistic gas storage valuation settings. A systematic comparison of polynomial regression, spline regression, and neural network approximations within a unified LSMC framework applied to a real storage operator has, to the authors' knowledge, not previously appeared in the literature. The present study addresses this gap by evaluating six approximation methods on two storage contracts, including the facilities of Gas Storage Denmark subject to EU seasonal filling obligations, and explicitly quantifying the economic cost of those regulatory constraints.

1.2 Approaches to gas storage valuation

This section provides an overview of different approaches to gas storage valuation outlined in Breslin et al. (2008), highlighting their underlying assumptions, strengths, and limitations. Intrinsic valuation methods, which rely on the forward curve to determine optimal injection and withdrawal strategies, are discussed first, followed by spot-based approaches that incorporate stochastic price dynamics, including Monte Carlo simulation and regression-based techniques.

1.2.1 Intrinsic valuation

Intrinsic valuation, also referred to as forward-based valuation, focusses on the deterministic component of gas prices, capturing seasonal price patterns embedded in the forward curve. The basic idea is to determine an optimal injection and withdrawal strategy based on observed forward prices, entering long positions to buy and inject gas when forward prices are relatively low, and entering short positions to sell and withdraw gas when forward prices are relatively high.

This approach does not explicitly model the stochastic behaviour of spot prices. Consequently, it yields a static operational strategy, as the intrinsic value is determined at the initial evaluation date. In practice, bid and ask forward curves should be used to reflect transaction costs and market liquidity, since relying on mid-market or closing prices may lead to an overestimation of intrinsic value. Although computationally efficient, this method typically underestimates the true storage value because it ignores the optionality that arises from price volatility.

1.2.2 Rolling intrinsic valuation

Rolling intrinsic valuation extends the intrinsic approach by allowing periodic updates to the forward curve and the associated operational strategy. At each decision time, the forward curve is re-evaluated, and the optimal strategy is adjusted accordingly. In practice, increases in intrinsic value are captured by entering new positions based on the updated forward curve, while existing positions are typically offset through opposite trades.

This approach captures some dynamic aspects of storage valuation by incorporating evolving market conditions. However, it still relies solely on the forward curve and does not fully account for the stochastic behaviour of spot prices. Consequently, although rolling intrinsic valuation improves upon the static intrinsic approach, it may still underestimate storage value, particularly in volatile markets such as the gas market. Furthermore, frequent re-evaluation and trading can lead to significant transaction costs, which should be considered in practical applications.

1.2.3 Spot-based valuation

Spot-based valuation methods incorporate the stochastic behaviour of gas prices rather than relying solely on deterministic forward prices. The objective is typically to maximise the expected total discounted revenue over the lifetime of the storage contract across all possible price paths. Under these approaches, spot price dynamics are modelled using stochastic processes, such as mean-reverting or jump-diffusion models, which are calibrated to the observed forward curve to ensure consistency with market prices. In the trivial case of zero volatility, spot-based valuation reduces to intrinsic valuation, yielding the same optimal operational strategy.

Numerical techniques are then applied to solve the resulting stochastic control problem. Common approaches include backward induction methods, such as binomial or trinomial trees, and regression-based methods such as the Least Squares Monte Carlo approach. In the latter, regression is used to approximate continuation values and identify optimal exercise strategies under uncertainty, while simulated price paths are generated to estimate the expected payoff of different operational strategies.

Spot-based methods may capture a greater share of the flexibility of the storage facility and therefore provide a more realistic valuation. They are typically computationally intensive, however, and they require the underlying price model to be carefully specified and calibrated.

Outline

The remainder of this thesis is organised as follows. Chapter 2 develops the spot price model, covering parameter estimation, calibration to the forward curve, and variance reduction techniques. Chapter 3 formulates gas storage valuation as a stochastic dynamic programming problem and develops the Least Squares Monte Carlo method for approximating continuation values under operational constraints. Chapter 4 introduces multi-layer perceptrons and Kolmogorov–Arnold Networks as continuation value approximators, covering their training procedures and regularisation strategies. Chapter 5 constructs the daily forward curve from discrete contract quotes and calibrates the simulated spot price process to market data. Chapter 6 presents valuation results for both the Gas Storage Denmark facility and the stylised Dutch salt cavern across all six approximation methods, and quantifies the economic cost of EU seasonal filling obligations. Chapter 7 discusses the comparative performance of the approximation methods, the economic interpretation of results, and the limitations of the modelling framework. Chapter 8 concludes on the findings.

2 | Spot price simulation

To model the stochastic behaviour of gas spot prices, we employ the one-factor model proposed by Schwartz (1997), which captures seasonality and mean reversion commonly observed in gas markets. In this model, the spot price $S(t)$ is assumed to follow the stochastic differential equation

$$\frac{\mathrm{d}S(t)}{S(t)} = \kappa[\mu(t) - \ln S(t)]\,\mathrm{d}t + \sigma\,\mathrm{d}B(t),$$

where $\mu(t)$ denotes the time-dependent mean-reversion level, κ the speed of mean reversion, σ the instantaneous volatility, and $B(t)$ a standard Brownian motion. Let $X(t) = \ln S(t)$, using Itô's formula, the dynamics of $X(t)$ are given by

$$\begin{aligned}
\mathrm{d}X(t) &= \mathrm{d}(\ln S(t)) \\
&= \frac{1}{S(t)}\,\mathrm{d}S(t) - \frac{1}{2}\frac{1}{S(t)^2}\,\mathrm{d}\langle S\rangle_t \\
&= \frac{1}{S(t)}\,\mathrm{d}S(t) - \frac{1}{2}\sigma^2\,\mathrm{d}t \\
&= \kappa\left[\mu(t) - \ln S(t)\right]\,\mathrm{d}t + \sigma\,\mathrm{d}B(t) - \frac{1}{2}\sigma^2\,\mathrm{d}t \\
&= \kappa\left[\mu(t) - X(t) - \frac{\sigma^2}{2\kappa}\right]\mathrm{d}t + \sigma\,\mathrm{d}B(t), \qquad (2.1)
\end{aligned}$$

where in the third step we used $\mathrm{d}\langle S\rangle_t = \sigma^2 S(t)^2\,\mathrm{d}t$, which follows from the quadratic variation of Brownian motion, $\mathrm{d}\langle B\rangle_t = \mathrm{d}t$. The form (2.1) corresponds to an Ornstein–Uhlenbeck process in logarithmic space.

The one-factor specification offers several computational advantages. The model admits a closed-form solution, its parameters can be estimated using standard linear methods, and Monte Carlo simulation is straightforward, which allows for efficient computation. Empirically, the model captures the main characteristics of gas spot prices that are most relevant for storage valuation, notably seasonality and strong mean reversion.

The model also has limitations, however. With only one state variable, it provides a simplified representation of spot-price dynamics and cannot capture features such as volatility clustering, sudden price spikes, or regime shifts. Nevertheless, for the purpose of gas storage valuation, the one-factor Schwartz model provides a workable balance between analytical tractability and empirical realism.

2.1 Parameter estimation and calibration

We next describe the estimation of the model parameters under the physical probability measure and the subsequent calibration of the risk-neutral drift required for valuation. Throughout, we work with the logarithmic spot price $X(t) = \ln S(t)$, whose dynamics are given in (2.1). The parameters governing short-term volatility and mean reversion are inferred from historical spot data, while the drift is calibrated to ensure consistency with observed forward prices.

Let $S(t_0), \ldots, S(t_N)$ denote spot price observations at equidistant times with step size Δt, and define $X_{t_i} = \ln S(t_i)$. For estimation purposes, it is convenient to absorb the constant term $\sigma^2/(2\kappa)$ in (2.1) into the mean function and consider the equivalent Ornstein–Uhlenbeck representation

$$\mathrm{d}X_t = \kappa\big(\tilde{\mu}(t) - X_t\big)\,\mathrm{d}t + \sigma\,\mathrm{d}B(t), \tag{2.2}$$

where $\tilde{\mu}(t) = \mu(t) - \sigma^2/(2\kappa)$ is the time-dependent long-term log-price mean, i.e., the level to which X_t reverts. To derive the discretisation, we multiply both sides of (2.2) by the integrating factor $e^{\kappa t}$. By Itô's product rule and substituting $\mathrm{d}X_t$ from (2.2), we obtain

$$\mathrm{d}\Big(e^{\kappa t}X_t\Big) = \kappa\tilde{\mu}(t)e^{\kappa t}\,\mathrm{d}t + \sigma e^{\kappa t}\,\mathrm{d}B(t),$$

which can now be integrated directly. Integrating from t_i to $t_{i+1} = t_i + \Delta t$ and solving for $X_{t_{i+1}}$ yields

$$X_{t_{i+1}} = e^{-\kappa\Delta t}X_{t_i} + \kappa e^{-\kappa t_{i+1}}\int_{t_i}^{t_{i+1}}\tilde{\mu}(s)e^{\kappa s}\,\mathrm{d}s + \sigma e^{-\kappa t_{i+1}}\int_{t_i}^{t_{i+1}}e^{\kappa s}\,\mathrm{d}B(s).$$

Assuming that $\tilde{\mu}(t)$ is constant on the interval $[t_i,t_{i+1}]$, i.e., $\tilde{\mu}(t) = \tilde{\mu}(t_i)$, the deterministic integral simplifies to

$$\kappa e^{-\kappa t_{i+1}}\int_{t_i}^{t_{i+1}}\tilde{\mu}(t_i)e^{\kappa s}\,\mathrm{d}s = \big(1 - e^{-\kappa\Delta t}\big)\tilde{\mu}(t_i).$$

The stochastic integral defines the innovation term

$$\varepsilon_{t_i} = \sigma e^{-\kappa t_{i+1}}\int_{t_i}^{t_{i+1}}e^{\kappa s}\,\mathrm{d}B(s).$$

Because this is an Itô integral with deterministic integrand, ε_{t_i} is Gaussian with mean zero. Its variance follows from the Itô isometry and the quadratic variation of Brownian motion:

$$\mathrm{Var}(\varepsilon_{t_i}) = \sigma^2 e^{-2\kappa t_{i+1}}\int_{t_i}^{t_{i+1}}e^{2\kappa s}\,\mathrm{d}s = \frac{\sigma^2}{2\kappa}\big(1 - e^{-2\kappa\Delta t}\big).$$

Hence,

$$X_{t_{i+1}} = e^{-\kappa\Delta t}X_{t_i} + \big(1 - e^{-\kappa\Delta t}\big)\tilde{\mu}(t_i) + \varepsilon_{t_i},$$

with

$$\varepsilon_{t_i} \sim \mathcal{N}\left(0, \frac{\sigma^2}{2\kappa}\left(1 - e^{-2\kappa\Delta t}\right)\right).$$

Moreover, the innovations are independent across time steps, since the stochastic integrals are taken over disjoint Brownian increments. This discretisation is exact for the Ornstein–Uhlenbeck process when the mean function $\tilde{\mu}(t)$ is piecewise constant on each interval $[t_i, t_{i+1}]$.

When the long-run mean is constant, the discretised process reduces exactly to an AR(1) representation. This reflects the well-known result that the Ornstein–Uhlenbeck process is the continuous-time analogue of an AR(1) process. In this case, the discretised dynamics can be written as

$$X_{t_{i+1}} = \alpha + \beta X_{t_i} + \varepsilon_{t_i},$$

where the coefficients satisfy $\beta = e^{-\kappa\Delta t}$ and $\alpha = (1 - \beta)\tilde{\mu}$.

In empirical applications to gas markets, however, the logarithmic spot price typically exhibits deterministic seasonal variation reflecting fluctuations in demand, particularly the higher consumption during winter months. Rather than removing this seasonal component prior to estimation, it is convenient to estimate the seasonal mean function and the mean-reversion parameter jointly. To this end, the long-run mean $\tilde{\mu}(t)$ is parameterised as a deterministic function of time, for example using sinusoidal terms or monthly dummy variables. Substituting this time-dependent mean into the discretised dynamics yields a regression specification in which the intercept becomes time dependent, while the autoregressive coefficient β remains unchanged. Estimating this equation by ordinary least squares allows the parameters governing the seasonal mean function and the autoregressive coefficient to be inferred jointly.

Denoting the ordinary least squares estimates of (α, β) by $(\hat{\alpha}, \hat{\beta})$, the continuous-time parameters are recovered as

$$\hat{\kappa} = -\frac{1}{\Delta t} \ln \hat{\beta}, \qquad \hat{\tilde{\mu}} = \frac{\hat{\alpha}}{1 - \hat{\beta}}. \tag{2.3}$$

Let $\hat{\sigma}_\varepsilon^2$ denote the empirical variance of the regression residuals. Matching this to the theoretical innovation variance implies

$$\hat{\sigma} = \sqrt{\hat{\sigma}_\varepsilon^2 \frac{2\hat{\kappa}}{1 - e^{-2\hat{\kappa}\Delta t}}}. \tag{2.4}$$

These estimates characterise the short-term volatility and mean-reversion properties of the spot price under the physical measure. In the constant-mean case, the formula for $\hat{\tilde{\mu}}$ applies directly. In the seasonal case, $\hat{\kappa}$ and $\hat{\sigma}$ are recovered identically using $\hat{\beta}$ and the residual variance, while the seasonal mean function $\hat{\tilde{\mu}}(t)$ is recovered coefficient-wise from the estimated time-dependent intercept terms divided by $(1 - \hat{\beta})$.

For valuation, the spot price must be simulated under a risk-neutral probability measure $\mathbb{Q}$. Under standard no-arbitrage assumptions, the existence of such a measure ensures that discounted asset prices are martingales and that derivative prices can be expressed as risk-neutral expectations. For diffusion models such as the log-price dynamics in (2.1), the change of measure from the physical measure $\mathbb{P}$ to the risk-neutral measure $\mathbb{Q}$ affects only the drift term of the process, while the diffusion coefficient remains unchanged.

In the original formulation of Schwartz (1997), the change of measure is parameterised via a constant market price of risk, which shifts the long-run mean by a constant amount λ, so that the risk-neutral long-run mean is $\tilde{\mu}^* = \tilde{\mu} - \lambda$. While this representation provides a clear economic interpretation of the risk premium, it is restrictive in applications where the forward curve exhibits pronounced seasonal and term-structure effects. In the present setting, we therefore allow for a time-dependent risk-neutral mean function $\tilde{\mu}_{\mathbb{Q}}(t)$, which is calibrated directly to the observed forward curve, rather than imposed parametrically.

Accordingly, we assume that the volatility and mean-reversion speed are invariant under the change of measure, so that $\kappa_{\mathbb{Q}} = \kappa$ and $\sigma_{\mathbb{Q}} = \sigma$, and absorb the entire market price of risk into the risk-neutral mean function $\tilde{\mu}_{\mathbb{Q}}(t)$. Under $\mathbb{Q}$, the log-price then evolves according to

$$\mathrm{d}X(t) = \kappa\big(\tilde{\mu}_{\mathbb{Q}}(t) - X(t)\big)\,\mathrm{d}t + \sigma\,\mathrm{d}B_{\mathbb{Q}}(t).$$

The theoretical foundation for this change of measure rests on the existence of an equivalent martingale measure, whose existence is guaranteed under the no-free-lunch-with-vanishing-risk condition of Delbaen and Schachermayer (1994). For diffusion models such as the one considered here, the change of measure can be constructed explicitly using Girsanov's theorem. Let $\theta(t)$ denote the market price of risk process and define the Radon–Nikodym density

$$\left.\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}\right|_{\mathcal{F}_t} = \exp\left(-\int_0^t \theta(s)\,\mathrm{d}B_{\mathbb{P}}(s) - \frac{1}{2}\int_0^t \theta(s)^2\,\mathrm{d}s\right).$$

Matching the drift shift required to move from $\tilde{\mu}(t)$ to $\tilde{\mu}_{\mathbb{Q}}(t)$, the market price of risk takes the explicit form

$$\theta(t) = \frac{\kappa(\tilde{\mu}(t) - \tilde{\mu}_{\mathbb{Q}}(t))}{\sigma}.$$

Since $\tilde{\mu}(t)$ and $\tilde{\mu}_{\mathbb{Q}}(t)$ are bounded deterministic functions, $\theta(t)$ is bounded. Thus, the Novikov condition

$$\mathbb{E}_{\mathbb{P}}\left[\exp\left(\frac{1}{2}\int_0^T \theta(s)^2 \,\mathrm{d}s\right)\right] < \infty$$

holds. Girsanov's theorem then implies that

$$B_{\mathbb{Q}}(t) = B_{\mathbb{P}}(t) + \int_0^t \theta(s)\,\mathrm{d}s$$

is a standard Brownian motion under $\mathbb{Q}$ (Glasserman, 2003). The spot price dynamics under $\mathbb{Q}$ are then obtained by substituting $\mathrm{d}B_{\mathbb{P}}(t) = \mathrm{d}B_{\mathbb{Q}}(t) - \theta(t)\,\mathrm{d}t$, which shifts the drift while the diffusion coefficient remains unchanged. Under the equivalent measure, discounted asset prices are martingales, so that the price of any attainable contingent claim with payoff H at maturity T is given by

$$V(0) = e^{-\delta T}\,\mathbb{E}_{\mathbb{Q}}[H]\,.$$

Here $\delta \geq 0$ denotes the continuously compounded risk-free rate. This expectation is approximated numerically using the Least Squares Monte Carlo algorithm, which is developed in Chapter 3.

Under the physical measure $\mathbb{P}$, the long-run mean function $\tilde{\mu}(t)$ is specified parametrically to capture deterministic seasonal patterns, as described above. Such specifications are estimated jointly with the mean-reversion speed and volatility using historical spot data and serve to separate predictable seasonal variation from stochastic price dynamics. For valuation purposes, however, the risk-neutral mean function $\tilde{\mu}_{\mathbb{Q}}(t)$ cannot be chosen freely and must be consistent with observed forward prices. Accordingly, while seasonal specifications of $\tilde{\mu}(t)$ are useful for identifying the physical dynamics of the spot price process, the risk-neutral drift is replaced by the forward-implied mean function to ensure arbitrage-free pricing.

Under $\mathbb{Q}$, the no-arbitrage forward price for delivery at time t is determined by the condition that entering a forward contract has zero initial cost. This requires that the present value of receiving $S(t)$ equals the present value of paying the fixed delivery price $F(0,t)$, giving

$$F(0,t) = \mathbb{E}_{\mathbb{Q}}[S(t) \mid S(0)]\,, \tag{2.5}$$

which serves as the calibration condition below. Since $X(t)$ is Gaussian under $\mathbb{Q}$ and $S(t) = \exp(X(t))$, this expectation takes the lognormal form

$$\mathbb{E}_{\mathbb{Q}}[S(t)] = \exp\left(m_{\mathbb{Q}}(t) + \frac{1}{2}\operatorname{Var}_{\mathbb{Q}}(X(t))\right),$$

where $m_{\mathbb{Q}}(t) = \mathbb{E}_{\mathbb{Q}}[X(t)]$ denotes the deterministic mean of the process. For the Ornstein–Uhlenbeck process, the conditional variance is given by

$$\operatorname{Var}_{\mathbb{Q}}(X(t)) = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa t}),$$

which depends only on the volatility and mean-reversion parameters. Imposing the equality (2.5) then determines the deterministic mean path $m_{\mathbb{Q}}(t)$ of the log-price process. Since $X(t)$ is Gaussian under $\mathbb{Q}$, inverting the lognormal relation gives

$$m_{\mathbb{Q}}(t) = \ln F(0,t) - \tfrac{1}{2}\operatorname{Var}_{\mathbb{Q}}(X(t)). \tag{2.6}$$

In simulation, it is convenient to decompose the log-price into a deterministic risk-neutral mean component and a stochastic deviation process,

$$X(t) = m_{\mathbb{Q}}(t) + Y(t), \tag{2.7}$$

where the residual $Y(t)$ follows the zero-mean Ornstein–Uhlenbeck process

$$\mathrm{d}Y(t) = -\kappa Y(t)\,\mathrm{d}t + \sigma\,\mathrm{d}B_{\mathbb{Q}}(t).$$

Since $m_{\mathbb{Q}}(0) = X(0)$, the residual starts at $Y(0) = 0$ and therefore has zero mean at all times, representing stochastic deviations from the deterministic mean path $m_{\mathbb{Q}}(t)$. By construction, this guarantees that $\mathbb{E}_{\mathbb{Q}}[S(t)] = F(0,t)$ for all maturities, which ensures arbitrage-free consistency with the forward market. The resulting spot price paths are subsequently used in the valuation of gas storage through simulation-based dynamic optimisation described in Chapter 3.

2.2 Variance reduction

Monte Carlo estimators may exhibit substantial sampling variability. Variance reduction techniques are therefore often employed to reduce the variance of the estimator without increasing the number of simulated paths. For a discussion of variance reduction techniques in Monte Carlo simulation, we refer the reader to Glasserman (2003).

To improve the efficiency of the Monte Carlo simulations, we employ antithetic sampling (Glasserman, 2003). This technique involves generating pairs of negatively correlated price paths by reversing the signs of the Brownian increments that drive the dynamics. In the spot price simulation in (2.1), the antithetic counterpart of a simulated path is obtained by replacing $\mathrm{d}B(t)$ with $-\,\mathrm{d}B(t)$. This construction aims to balance unusually large or small outputs from one path with the value computed from the antithetic path, reducing variance when the two outputs are negatively correlated.

To examine this type of variance reduction more formally, consider a pair of outputs $(Y_i, \tilde{Y}_i)$ generated from a simulated path and its corresponding antithetic counterpart. Since Y_i and $\tilde{Y}_i$ have the same distribution, their variances coincide, and it follows that

$$\operatorname{Var}\left[Y_i + \tilde{Y}_i\right] = 2\operatorname{Var}\left[Y_i\right] + 2\operatorname{Cov}\left[Y_i, \tilde{Y}_i\right].$$

If these two outputs were instead generated from independent simulations, the covariance term would be zero, so the variance of the sum would simply be $2\operatorname{Var}\left[Y_i\right]$. Antithetic sampling therefore achieves variance reduction precisely when

$$\operatorname{Cov}\left[Y_i, \tilde{Y}_i\right] < 0.$$

A sufficient condition ensuring negative covariance is that the simulation algorithm defines a monotone mapping from its input variables to the resulting output (Esary et al., 1967). In the storage valuation algorithm described in Chapter 3, however, the mapping from Brownian increments to the final value estimate is not necessarily strictly monotone. The use of regression to estimate continuation values and the presence of discrete control decisions may introduce non-monotonicities, since small perturbations of a simulated price path can alter the optimal injection or withdrawal decision. Nevertheless, because the simulated price paths themselves depend monotonically on the Brownian increments and the storage value generally increases with higher price levels, antithetic paths tend to produce negatively correlated outputs in practice, thereby reducing the variance of the resulting value estimates.

In the storage valuation problem considered in this thesis, a substantial part of the variability of the value estimate arises not from the randomness of the simulated price paths but from the estimation and optimisation steps of the algorithm. The value is obtained through a backward dynamic optimisation procedure with discrete operational decisions, so small perturbations of simulated price paths may lead to different actions being selected along the path, and thus affect the resulting value estimate. Since these sources of variability are not directly addressed by standard variance reduction techniques for Monte Carlo sampling, we restrict attention to antithetic sampling, which is simple to implement and can reduce variance in the price path simulation without adding further complexity to the algorithm.

3 | Least Squares Monte Carlo approach

In this chapter, we develop the theory of gas storage valuation using the Least Squares Monte Carlo (LSMC) method proposed by Boogert and De Jong (2007). This method is a generalisation of the least squares Monte Carlo algorithm for valuation of American options introduced by Longstaff and Schwartz (2001). In the following, we consider a financial storage contract from the holder's perspective. This contract can be applicable to both real physical storage facilities and financial storage flexibility contracts. The holder of the contract is faced with the problem of optimally timing gas injections and withdrawals in order to maximise expected future payoffs over the contract duration. This problem is closely related to that of an American option, where the holder must decide when and if to exercise the option. The storage contract generalises the American option in several respects. The holder must choose among multiple possible actions at each point in time, rather than a single exercise decision, and the optimisation is typically subject to volumetric constraints on injection, withdrawal, and capacity.

3.1 Storage contract specification

We assume that the contract is initiated at time $t = 0$ and settled at time $t = T + 1$. Actions may be made at each discrete time point $t = 1, \dots, T$ after the spot price $S(t)$ is revealed at the start of the time point. For simplicity, time is measured in days. An action defines either injection of gas, withdrawal of gas, or the decision to remain inactive, all subject to volumetric constraints. We denote an action on day t by a volume change $\Delta v(t)$, where $\Delta v(t) > 0$ represents injection, and $\Delta v(t) < 0$ represents withdrawal. Since the first action can occur on day $t = 1$, it is natural to define $\Delta v(0) \coloneqq 0$. For simplicity, it is assumed that the stored gas volume remains constant over time in the absence of any action, and that no gas is lost through leakage or other physical effects. Throughout, it is assumed that the prevailing forward curve fully reflects the market's expectations regarding the optimal use of storage, and that the holder's trading activity has no impact on market prices. We denote by $v(t)$ the storage volume at the start of day t. Thus, an action taken on day 1 results in a volume $v(2)$ at the end of day 1, which also serves as the volume at the start of day 2. Since $\Delta v(t) = v(t+1) - v(t)$, the accumulated volume on day t can be expressed in terms of $v(0)$ by

$$v(t) = v(0) + \sum_{i=1}^{t} \Delta v(i-1).$$

Gas injection, corresponding to positive volume changes, that must be bought at the market, represents a cost, whereas gas withdrawal, corresponding to negative volume changes, represents a profit. We define the payoff on day t by

$$G(S(t), \Delta v(t)) \coloneqq \begin{cases} -c(S(t))\Delta v(t) & \text{injection on day } t, \\ 0 & \text{no action on day } t, \\ -p(S(t))\Delta v(t) & \text{withdrawal on day } t, \end{cases}$$

for $t = 0, \dots, T$. Here, $c(S(t))$ and $p(S(t))$ represent the unit cost of injection and unit profit of withdrawal, respectively, defined by

$$\begin{aligned} c(S(t)) &\coloneqq (1+a_1)S(t)+b_1, \\ p(S(t)) &\coloneqq (1-a_2)S(t)-b_2, \end{aligned} \tag{3.1}$$

where $a_1, a_2 \in \mathbb{R}_0^+$ represent transaction costs, and $b_1, b_2 \in \mathbb{R}_0^+$ represent bid-ask spreads, satisfying $c(S(t)), p(S(t)) \geq 0$. By construction, the payoff $G(S(t), \Delta v(t))$ and the volume change $\Delta v(t)$ have opposite signs. At the settlement date $T+1$, the holder receives a potential penalty, denoted by $q(S(T+1), v(T+1))$, which may depend on remaining gas in the storage and other contractual factors.

To incorporate physical requirements, we impose volumetric constraints. First, the storage volume must remain between a minimum level $v^{\min}(t) \geq 0$ and a maximum level $v^{\max}(t) \geq 0$ for $t = 1, \ldots, T+1$:

$$v^{\min}(t) \leq v(t) \leq v^{\max}(t).$$

The time-dependent nature of these volume restrictions may, for instance, serve to prevent rock deformations or maintain adequate reservoir pressure by requiring a higher minimum storage level during certain periods of the year, while seasonal factors such as scheduled maintenance or regulatory obligations to ensure winter supply security can also impose time-varying constraints (Boogert and De Jong, 2007).

Second, we restrict the allowed daily volume injection rate $i^{\max}(t, v(t)) \geq 0$ and withdrawal rate $w^{\max}(t, v(t)) \geq 0$ by requiring the daily volume change to satisfy

$$-w^{\max}(t, v(t)) \leq \Delta v(t) \leq i^{\max}(t, v(t)), \tag{3.2}$$

for $t = 1, \ldots, T+1$. For most storage volumes, these limits are constant, except at particular high or low levels, where physical constraints make injection or withdrawal more difficult.

Lastly, we define the set of all admissible volume levels on day t as

$$\mathcal{V}(t) \coloneqq \{v \mid v^{\min}(t) \leq v \leq v^{\max}(t)\},$$

and the set of allowed actions on day t at volume $v(t)$ as

$$\mathcal{D}(t, v(t)) \coloneqq \{\Delta v \mid v(t) + \Delta v \in \mathcal{V}(t+1), -w^{\max}(t, v(t)) \leq \Delta v \leq i^{\max}(t, v(t))\}.$$

3.2 Valuation framework

The value of the storage contract is the expected value of the accumulated future payoffs $G(S(t), \Delta v(t))$, using the optimal strategy $\pi \coloneqq \{\pi(1, S(1), v(1)), \ldots, \pi(T, S(T), v(T))\}$ where $\pi(t, S(t), v(t))$ represents the optimal volume change on day t. We denote by $V(t, S(t), v(t))$ the value of a storage contract initiated at the start of the day t at volume level $v(t)$ and terminated on day $T+1$, defined by

$$\begin{aligned} V(t, S(t), v(t)) \coloneqq \sup_{\{\Delta v(k) \in \mathcal{D}(k, v(k))\}_{k=t}^{T}} \mathbb{E}_{\mathbb{Q}} \Bigg[& \sum_{k=t}^{T} e^{-\delta(k-t)} G(S(k), \Delta v(k)) \\ & + e^{-\delta(T+1-t)} q(S(T+1), v(T+1)) \mid S(t), v(t) \Bigg], \end{aligned} \tag{3.3}$$

where $\delta \in \mathbb{R}_0^+$ denotes the continuously compounded risk-free rate, and where the expectation is taken with respect to a risk-neutral pricing measure $\mathbb{Q}$. The objective is thus to determine the optimal decision rule for all t and all admissible volume levels $v(t)$.

The continuation value on day t is defined as the expected discounted value of the contract after taking an admissible action $\Delta v \in \mathcal{D}(t, v(t))$,

$$C(t, S(t), v(t), \Delta v(t)) := \mathbb{E}_{\mathbb{Q}}\left[e^{-\delta} V(t+1, S(t+1), v(t) + \Delta v(t)) \mid S(t), v(t)\right], \tag{3.4}$$

for $t = 0, \ldots, T$. We can now define the value of a storage contract recursively by the following dynamic program:

$$V(T+1, S(T+1), v(T+1)) = q(S(T+1), v(T+1)), \tag{3.5}$$

$$V(t, S(t), v(t)) = \max_{\Delta v \in \mathcal{D}(t, v(t))} \{G(S(t), \Delta v) + C(t, S(t), v(t), \Delta v)\}, \tag{3.6}$$

for $t = 0, \ldots, T$. Thus, the value function consists of two components, namely the immediate payoff from the current action and the continuation value, which reflects expected future payoffs. Equations (3.5)–(3.6) therefore constitute a discrete-time Bellman equation, where the optimal policy is obtained by maximising over all admissible actions. To solve the problem in (3.5)–(3.6), we must approximate the continuation values (3.4) for all t and all allowed actions $\Delta v \in \mathcal{D}(t, v(t))$ for admissible volume levels $v(t) \in \mathcal{V}(t)$.

3.3 Least Squares Monte Carlo method

In the original method by Longstaff and Schwartz (2001), the continuation values are approximated by

$$\hat{C}_Q(t, S(t), v(t), \Delta v(t)) := \sum_{q=1}^{Q} \phi_q\left(t, S(t), v(t), \Delta v(t)\right) \beta_{q,t}, \tag{3.7}$$

for regression coefficients $\beta_{q,t} \in \mathbb{R}$ and a finite number of basis functions $Q \in \mathbb{N}$. This approximation is based on the assumption that the conditional expectation in (3.4) can be represented as a linear combination of a countable set of basis functions. The coefficients $\beta_{q,t}$ are estimated by a sequence of least-squares regressions of the realised cash flows, which serve as proxies for the true continuation values, on the current state variables. The continuation value is then approximated by substituting the estimated regression coefficients. From this, a decision rule is determined. For notational simplicity, we omit the superscript Q in what follows when referring to estimates of continuation values.

3.3.1 Dimensionality reduction and discretisation

We first reduce the dimensionality of the problem so that estimation of the continuation value $C(t, S(t), v(t), \Delta v(t))$ in (3.4) depends solely on the current spot price $S(t)$. Note first that the continuation value depends only on the accumulated volume and not simultaneously on the volume level $v(t)$ and the volume change Δv. To see this, consider a case where the storage has a volume level l and the action $\Delta v = 0$ is taken, thus requiring an approximation of $C(t, S(t), l, 0)$. If instead the storage volume were $l - \alpha$ and the action $\Delta v = \alpha$ were taken, both cases would result in a volume level of l at time $t + 1$. This implies that $C(t, S(t), l, 0) = C(t, S(t), l - \alpha, \alpha)$. Hence, continuation values can be written as

$$C(t, S(t), v(t), \Delta v(t)) := C(t, S(t), v(t) + \Delta v(t)) = C(t, S(t), v(t+1)).$$

Following the approach in Longstaff and Schwartz (2001), the time variable is omitted, as separate regressions are carried out for each time step. Additionally, since the relationship between volume level, price, and continuation value is often not very smooth (Boogert and De Jong, 2007), a separate regression is conducted for each volume level by discretising the range of storage volume into $N-1$ intervals of equal width α, after which the regression parameters are estimated individually for each volume point $v(t+1;n) := (n-1)\alpha$, $n = 1, \ldots, N$. Note that not all volume points are necessarily attainable at each time t, depending on the volumetric constraints. Based on these considerations, the basis functions ϕ_q in (3.7) now depend solely on the spot price $S(t)$.

3.3.2 Estimation and regression

We can now formulate an estimate for the continuation values. We select $M \in \mathbb{N}$ and simulate M independent price paths $(S^{(i)}(0), \ldots, S^{(i)}(T+1))$, for $i = 1, \ldots, M$, where $(S^{(1)}(0) = \cdots = S^{(M)}(0))$. The continuation value $C^{(i)}$ along the price path i is then approximated by

$$C^{(i)}(t, S^{(i)}(t), v(t+1;n)) \approx e^{-\delta} Y^{(i)}(t+1, S^{(i)}(t+1), v(t+1;n)), \tag{3.8}$$

where $Y^{(i)}(t+1, S^{(i)}(t+1), v(t+1;n))$ denotes the accumulated value of future realised cash flows in path i following optimal decisions starting at time $t+1$ with volume level $v(t+1)$ and price $S(t+1)$. We remark that $Y^{(i)}$ denotes actual realised cash flows, not the estimated continuation value $\hat{C}^{(i)}$. As noted by Longstaff and Schwartz (2001), using $\hat{C}^{(i)}$ directly would introduce an upward bias, since the convexity of the maximum operator in (3.6) means that measurement error in $\hat{C}^{(i)}$ inflates the estimated value. The right-hand side of (3.8) is estimated using ordinary least squares, yielding coefficients $\hat{\beta}_{q,t}$. These coefficients are then substituted into (3.7) to obtain an approximation $\hat{C}^{(i)}(t, S^{(i)}(t), v(t+1;n))$ of the true continuation value, for all volume points $v(t;n)$.

Note that in (3.8), we need only estimate continuation values that are at attainable volume points, that is, $v(t+1;n) \in \mathcal{V}(t+1)$. This is because the decision rule in (3.6) only requires continuation values at attainable volume levels. However, due to the discretisation of the attainable volume levels, and consequently the admissible actions, a very fine grid of volume points may be required to obtain accurate estimates of continuation values. To alleviate this, interpolation between adjacent volume points can be employed in order to approximate continuation values for intermediate volume levels. Specifically, the continuation value at a specified intermediate volume level $v(t+1)$ is computed as a distance-weighted average of the continuation values at the two adjacent volume points, $v(t+1;n^*)$ and $v(t+1;n^*+1)$, where

$$n^* := \sup \{n \in \{1, \ldots, N\} \mid (n-1)\alpha \leq v(t+1)\}.$$

That is,

$$\hat{C}(t, S(t), v(t+1)) := (1-w_{n^*})\hat{C}(t, S(t), v(t+1;n^*)) + w_{n^*}\hat{C}(t, S(t), v(t+1;n^*+1)), \tag{3.9}$$

with the interpolation weight defined as

$$w_{n^*} := \frac{v(t+1) - v(t+1;n^*)}{\alpha}.$$

This interpolation allows us to use a coarser grid while reducing discretisation error in the continuation value approximations. The interpolation is particularly useful when the attainable volume levels are time-dependent and cannot be expressed as exact multiples of the injection and withdrawal rates, which is typically the case in practice.

With these approximations of the continuation values, we can determine a pathwise decision rule $\hat{\pi}^{(i)}(t, S^{(i)}(t), v(t))$ defined by

$$\hat{\pi}^{(i)}(t, S^{(i)}(t), v(t)) := \underset{\Delta v \in \mathcal{D}(t, v(t))}{\arg\max} \left\{ G(S^{(i)}(t), \Delta v) + \hat{C}^{(i)}(t, S^{(i)}(t), v(t) + \Delta v) \right\}, \tag{3.10}$$

and a corresponding approximation of the value of future accumulated cash flows:

$$\begin{aligned} \hat{Y}^{(i)}(t+1, S^{(i)}(t+1), v(t+1)) := G(S^{(i)}(t+1), \hat{\pi}^{(i)}(t+1, S^{(i)}(t+1), v(t+1))) \\ + e^{-\delta} \hat{Y}^{(i)} \Big(t+2, S^{(i)}(t+2), v(t+2) \\ + \hat{\pi}^{(i)}(t+1, S^{(i)}(t+1), v(t+1)) \Big). \end{aligned} \tag{3.11}$$

This backward-induction procedure is applied sequentially in time and across all discretised volume levels. The procedure starts at $T+1$, where, according to (3.5), the continuation value equals the penalty function, defined as

$$Y^{(i)}(T+1, S^{(i)}(T+1), v(T+1)) = q(S^{(i)}(T+1), v(T+1)),$$

for $i = 1, \dots, M$. Finally, the estimated value of the storage contract is obtained as the average of the simulated accumulated cash flows over all simulated paths by defining

$$\hat{V}(0, S(0), v(0)) = \frac{1}{M} \sum_{i=1}^{M} \hat{Y}^{(i)}(1, S^{(i)}(1), v(0)). \tag{3.12}$$

We note that unrolling the recursion in (3.11) from the terminal condition (3.5) gives

$$\hat{Y}^{(i)}(1, \cdot) = \sum_{s=1}^{T} e^{-\delta(s-1)} G\Big(S^{(i)}(s), \hat{\pi}^{(i)}(s, S^{(i)}(s), v(s))\Big) + e^{-\delta T} q\Big(S^{(i)}(T+1), v^{(i)}(T+1)\Big).$$

Hence, $\hat{Y}^{(i)}(1, \cdot)$ already equals the time-0 present value of all future cash flows, with the first payoff carrying discount factor $e^0 = 1$. No additional discounting is therefore required in (3.12).

An algorithm for approximating the value of the storage contract is formulated in Algorithm 1.

Algorithm 1: Least squares Monte Carlo

```
input  : Number of simulated paths $M$, interest rate $\delta$, time horizon $T+1$,
         collection of simulated paths $S_0^{(i)}, \dots, S_{T+1}^{(i)}$ for $i = 1, \dots, M$, width of
         volume points $\alpha \in \mathbb{R}_0^+$ and corresponding number of volume points
         $N = \frac{\sup_t \mathcal{V}(t)}{\alpha}$.
output : Estimates $\hat{C}$ of continuation values and estimate $\hat{V}(0)$ of the value of the
         storage contract.
Set $\hat{Y}^{(i)}(T+1, S^{(i)}(T+1), v(T+1;n)) := q(S^{(i)}(T+1), v(T+1;n))$, for
  $i = 1, \dots, M$ and all allowed volume points $v(T+1;n) \in \mathcal{V}(T+1)$, $n = 1, \dots, N$.
for $t = T, \dots, 1$ do
    for $n = 1, \dots, N$ do
        if $v(t+1;n) \in \mathcal{V}(t+1)$ then
            Run ordinary least squares regression on $e^{-\delta}\hat{Y}(t+1, S(t+1), v(t+1;n))$
              according to (3.7) to obtain an approximation
              $\hat{C}^{(i)}(t, S^{(i)}(t), v(t+1;n))$, for $i = 1, \dots, M$.
        end if
        For intermediate attainable volume levels $v(t+1) \in \mathcal{V}(t+1)$, such that
          $v(t+1;n) \le v(t+1) \le v(t+1;n+1)$, set
              $\hat{C}^{(i)}(t, S^{(i)}(t), v(t+1)) = (1-w_n)\hat{C}^{(i)}(t, S(t), v(t+1;n))$
                                                  $+ w_n\hat{C}^{(i)}(t, S(t), v(t+1;n+1)),$
          where $w_n = \frac{v(t+1) - v(t+1;n)}{\alpha}$, according to (3.9).
        Use the approximated continuation values to determine whether to update
          the decision rule $\hat{\pi}^{(i)}(t, S^{(i)}(t), v(t))$ according to (3.10), for $i = 1, \dots, M$,
          for the volume point $v(t;n)$ and intermediate attainable volume levels $v(t)$.
    end for
    Calculate the estimated accumulated value of future cash flows
      $\hat{Y}^{(i)}(t, S^{(i)}(t), v(t))$ according to (3.11), for $i = 1, \dots, M$.
end for
Set $\hat{V}(0, S(0), v(0)) := \frac{1}{M}\sum_{i=1}^{M} \hat{Y}^{(i)}(1, S^{(i)}(1), v(0))$.
```

3.4 Convergence properties

The estimate $\hat{V}$ derived from the Least Squares Monte Carlo algorithm approximates the true storage value function V. Since V is defined in (3.3) as the supremum over all admissible strategies, any specific decision rule, including the one produced by Algorithm 1, yields an expected value no greater than V. Consequently, the value of the policy obtained from LSMC constitutes a lower bound for the true value function, as stated in the following proposition.

Proposition 3.1

For any $Q, T \in \mathbb{N}$ and $t < T$, where Q is the number of basis functions, and T is the time horizon, assume that the functional form of $\hat{C}(t, S^{(i)}(t), v(t+1))$ is the same for all $i \in \mathbb{N}$. Then the following inequality holds $\mathbb{P}$-a.s.:

$$V(t, S(t), v(t)) \ge \lim_{M\to\infty} \frac{1}{M}\sum_{i=1}^{M} \hat{Y}^{(i)}(t, S^{(i)}(t), v(t)).$$

Proof: We present an outline of the proof. A more formal version can be found in the appendix of Longstaff and Schwartz (2001). Clearly, $V(t, S(t), v(t)) \geq \mathbb{E}[\hat{Y}(t, S(t), v(t))]$ for any admissible decision rule, by (3.3), and since the functional form of $\hat{C}(t, S^{(i)}(t), v(t+1))$ is the same across all paths, the accumulated cash flows $\hat{Y}^{(i)}(t, S^{(i)}(t), v(t))$ are independent and identically distributed. Then the strong law of large numbers implies that

$$\mathbb{P}\left[\lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^{M} \hat{Y}^{(i)}(t, S^{(i)}(t), v(t)) = \mathbb{E}\left[\hat{Y}(t, S(t), v(t))\right]\right] = 1,$$

which completes the proof. ■

In the context of American options, Clément et al. (2002) showed that the LSMC estimator converges to the true value as both the number of simulated paths M and the number of basis functions Q tend to infinity, and that the normalised Monte Carlo error is asymptotically Gaussian. Their analysis relies on the structure of the optimal stopping problem. In this setting, the state variable is the exogenous Markov process $S(t)$, and the continuation value at time t belongs to the Hilbert space $L^2(\sigma(S(t)))$. Clément et al. (2002) assumes that the family of basis functions is total in this space, which ensures that the conditional expectation of the value function can be approximated arbitrarily well as $Q \to \infty$. This density condition is what allows the regression operator to converge to the true conditional expectation operator.

In the gas storage problem, the structure of the dynamic optimisation problem differs from that of optimal stopping. The state is given by the pair $(S(t), v(t))$, where the volume level $v(t)$ is endogenous and evolves according to the chosen control $\Delta v(t)$. Consequently, the continuation value takes the form $C(t, S(t), v(t+1))$ and is an element of $L^2(\sigma(S(t), v(t)))$. In the implementations of the LSMC storage algorithm in Boogert and De Jong (2007), the inventory is discretised, and the regression step is performed separately for each admissible volume level. For a fixed volume level v, the regression therefore approximates the function $C_v(t, S) = C(t, S, v)$, which belongs to $L^2(\sigma(S(t)))$. In this case, the density condition on the basis functions may still hold for each individual regression. Nevertheless, the convergence result of Clément et al. (2002) does not directly apply, since their analysis relies on the binary optimal stopping structure, whereas gas storage valuation leads to a stochastic control problem with multiple admissible actions and additional discretisation of the inventory state.

Although one could in principle approximate the continuation value by employing basis functions depending jointly on price and volume, empirical evidence indicates that this approach performs poorly in practice. In particular, Boogert and De Jong (2007) report that, across a broad range of tested storage problems, the relationship between volume, price, and continuation value is often insufficiently smooth for reliable regression, with unsatisfactory convergence results even for high-order polynomial bases in (S, v). One explanation is that extending the set of basis functions tends to increase multicollinearity, which inflates the variance of the estimated coefficients and destabilises the regression. Moreover, the operational constraints on admissible volume levels may introduce kinks or discontinuities in the continuation value as a function of volume, further complicating the regression task.

Furthermore, the inventory is discretised into grid points $v(t; n)$ and separate regressions are performed for each grid level, typically with interpolation applied between them, as described in Subsections 3.3.1 and 3.3.2. This results in a collection of regression problems indexed by the inventory level, where each regression approximates the continuation value as a function of the current price. For a fixed v, the regression can still be interpreted as an L^2-projection onto a finite-dimensional subspace of $L^2(\sigma(S(t)))$. However, the overall valuation algorithm combines these projections with interpolation across inventory levels and with a discrete optimisation over admissible control actions.

These additional approximation steps alter the structure of the backward dynamic programming recursion relative to the optimal stopping problem considered in Clément et al. (2002). In particular, the presence of endogenous state dynamics, inventory discretisation, and interpolation between grid points introduces sources of approximation error that are not covered by their convergence analysis. To the best of our knowledge, an analogous convergence theorem for LSMC methods applied to controlled Markov decision problems such as gas storage valuation has not yet been established.

Despite the lack of theoretical convergence results, Boogert and De Jong (2007) provide empirical evidence supporting the convergence of the LSMC method for gas storage valuation. From a variety of test cases, they observe that the estimated storage value stabilises quickly as the number of simulated paths M increases, using only a modest number of basis functions. As observed by Longstaff and Schwartz (2001), empirical results suggest that the model is generally robust to the choice of basis functions. This suggests that, while theoretical convergence guarantees are lacking, the LSMC method remains a practical and effective tool for gas storage valuation in many scenarios.

These observations highlight both the theoretical limitations and the practical challenges of the classical regression-based approach. To address these issues, we propose in Chapter 6 to replace the regression step in the LSMC algorithm with a neural-network approximation of the continuation value. Neural networks provide a flexible, nonparametric alternative to fixed basis functions and may be more capable of capturing the highly non-linear and locally irregular dependence of the continuation value on (S, v) induced by operational constraints. By learning the continuation value directly from simulated data, this approach aims to mitigate the instability and approximation bias associated with polynomial regressions and thereby improve the numerical performance of the valuation procedure.

3.5 In-sample and out-of-sample valuation

A practical consideration in the LSMC method concerns the distinction between in-sample and out-of-sample valuation. In-sample valuation uses the same simulated price paths both to estimate the continuation values and to evaluate the resulting policy. In contrast, the out-of-sample estimate uses the same decision rule but evaluates it on a new set of simulated price paths. The two estimators are defined as

$$V_{\text{in}} \coloneqq \hat{V}(0, S(0), v(0)) = \frac{1}{M}\sum_{i=1}^{M} \hat{Y}^{(i)}(1, S^{(i)}(1), v(0)),$$

$$V_{\text{out}} \coloneqq \frac{1}{M}\sum_{i=1}^{M}\left[\sum_{t=1}^{T} e^{-\delta(t-1)} G\Big(S^{(i)}(t), \hat{\pi}(t, S^{(i)}(t), v(t))\Big) + e^{-\delta T} q\Big(S^{(i)}(T+1), v^{(i)}(T+1)\Big)\right].$$

Thus, the out-of-sample valuation procedure involves both a backward induction step, in which the decision rule is estimated, and a forward simulation step, in which this rule is evaluated on new price paths.

A well-performing algorithm should yield closely matching in-sample and out-of-sample valuations. Both Longstaff and Schwartz (2001) and Boogert and De Jong (2007) report that, in their numerical experiments, the in-sample and out-of-sample estimates are nearly identical, indicating that the LSMC method effectively captures the essential features of the storage valuation problem without materially overfitting to the specific simulated paths used for estimation.

From a computational perspective, the two approaches differ substantially. Out-of-sample valuation requires storing the decision rule for all time steps and all volume points during the backward induction, which amounts to a three-dimensional array of regression parameters. This significantly increases both memory usage and computational time. In contrast, an in-sample valuation requires only storing the regression parameters for a single volume grid point at a time, and once that grid point has been processed, the corresponding parameters can be discarded, which improves the computational efficiency. The limitation is that in-sample valuation does not provide information about the distribution of realised volume paths. For valuation purposes, this is often of limited concern, whereas for operational analysis of physical storage it may be relevant. In the latter case, one may combine an in-sample valuation, based on a large number of paths, with a smaller out-of-sample run (Boogert and De Jong, 2007).

4 | Neural networks approach

Neural networks are, in essence, non-linear statistical models used in regression or classification tasks. For classical feedforward neural networks, also known as multi-layer perceptrons, the idea is to construct derived features through learned linear combinations of the input variables and subsequently model the target variable by applying a non-linear activation function to these features. This combination of a linear feature construction and a non-linear transformation constitutes the characteristic two-stage structure of neural networks.

In contrast to regression-based approaches relying on fixed basis functions, this architecture allows the model to adapt its functional form to the data rather than requiring the functional structure to be specified a priori. As a result, neural networks may approximate complex, highly non-linear relationships and accommodate local irregularities more effectively than polynomial approximations. These properties motivate their use in the gas storage valuation problem considered in this project, where operational constraints can induce non-smooth continuation values and classical LSMC regressions may not provide stable or accurate estimates, as discussed in Section 3.4.

The framework of classical neural networks is inspired by the structure and function of the human brain. Just as the brain consists of neurons connected by synapses, neural networks are composed of interconnected nodes (neurons) that transmit signals to each other. Each connection carries a weight representing the strength of the signal, and learning occurs by adjusting these weights based on experience, which is analogous to how the brain strengthens or weakens synaptic connections.

In this chapter, we explore the single-layer perceptron, which illustrates how inputs are linearly combined and transformed through an activation function. We then extend this to multi-layer perceptrons, where additional layers are included in the model. Finally, we consider a relatively recent type of neural networks, namely the Kolmogorov–Arnold Network, whose structure and composition differ fundamentally from the classical architectures.

4.1 Single- and multi-layer perceptrons

We first examine the single-layer perceptron (SLP) which is illustrated in the network diagram in Figure 4.1. This network can be used for both regression and classification. For regression problems, usually $K = 1$, corresponding to a single response variable. However, since the framework is capable of handling multiple quantitative responses, we discuss the general case.

We consider an input vector $x = (x_1, \dots, x_p) \in \mathbb{R}^p$ and seek to model a response variable $Y = (Y_1, \dots, Y_K)$, either quantitative or qualitatively. For a K-class classification problem with mutually exclusive classes, the target variable is represented by a vector of K dummy variables, containing a one in the position corresponding to the true class and zeros elsewhere. This is known as 'one-hot encoding'. In the output layer of the model in Figure 4.1, Y_k denotes the estimated probability that the observation belongs to class k, for $k = 1, \dots, K$. As mentioned above, the objective is to create derived features Z_m, $m = 1, \dots, M$, defined as linear combinations of the inputs, and model the target variable

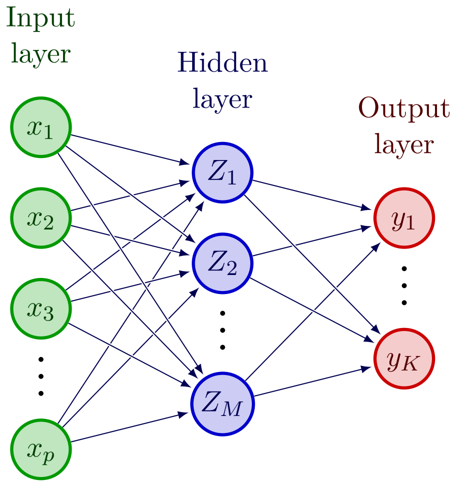


Figure 4.1: Single-layer perceptron neural network.

Y as a non-linear function of these features. Formally,

$$\begin{aligned} Z_m &= \sigma(\alpha_{m0} + \alpha_m^\top X), \quad m = 1, \dots, M, \\ T_k &= \beta_{k0} + \beta_k^\top Z, \quad k = 1, \dots, K, \\ f_k(X) &= g_k(T), \quad k = 1, \dots, K, \end{aligned} \tag{4.1}$$

where $Z = (Z_1, \dots, Z_M)$, and $T = (T_1, \dots, T_K)$, and where the activation function σ is prespecified (Hastie et al., 2009). Traditionally, the activation function is the sigmoid, $\sigma(z) = 1/(1 + e^{-z})$. Alternative choices include the ReLU (rectified linear unit), $\sigma(z) = \max(0, z)$, as well as the sign and hyperbolic tangent functions. These functions are displayed in Figure A.1. The ReLU offers improved computational efficiency compared with the sigmoid function and promotes sparse representations by driving some activations to zero, thereby effectively deactivating certain units in the network. It is also less prone to vanishing gradient problems than sigmoid activations, whose derivatives can become very small for large positive or negative inputs during backpropagation, thereby slowing learning. A potential drawback, however, is the risk of “dead neurons”, where a unit becomes permanently inactive and outputs zero for all inputs, which may hinder learning.

The intermediate units Z_m, referred to as hidden units since their values are not directly observable, serve as a non-linear basis expansion of the original inputs X. The central aspect of neural networks is that the parameters of the basis functions are learned from the data.

For regression problems, the final transformation function g_k is usually taken to be the identity function, i.e., $g_k(T) = T_k$, whereas for K-class classification, the standard choice is the softmax function

$$g_k(T) = \frac{\exp(T_k)}{\sum_{\ell=1}^{K} \exp(T_\ell)}, \tag{4.2}$$

which ensures strictly positive outputs that sum to one. Note that if the activation function σ is chosen to be the identity, the model collapses to a linear model.

An important theoretical result in the context of neural networks is the Universal Approximation Theorem (UAT), which describes the representational capability of feed-forward networks (Hornik et al., 1989). It states that a neural network with one hidden layer as defined above and a suitable non-linear activation function can approximate any continuous function on a compact subset of $\mathbb{R}^p$ to arbitrary accuracy, given a sufficient number of hidden units.

> **Theorem 4.1 (Universal Approximation Theorem)**
> Let $g : \mathbb{R} \to \mathbb{R}$ be a continuous, non-polynomial function, and let $K \subseteq \mathbb{R}^p$ be a compact set. Then, for any continuous function $f : K \to \mathbb{R}$ and any $\varepsilon > 0$, there exist $N(\varepsilon) \in \mathbb{N}$ and parameters
>
> $$w_n \in \mathbb{R}^p, \quad w_{n0}, \beta_0, \beta_n \in \mathbb{R}, \quad n = 1, \dots, N(\varepsilon),$$
>
> such that the function
>
> $$h_\varepsilon(x) = \beta_0 + \sum_{n=1}^{N(\varepsilon)} \beta_n\, g(w_n^\top x + w_{n0})$$
>
> satisfies
>
> $$\sup_{x \in K} |f(x) - h_\varepsilon(x)| < \varepsilon.$$

The single-layer perceptron framework can be extended to a multi-layer perceptron (MLP) by introducing additional hidden layers, each performing non-linear transformations of the outputs from the preceding layer. Specifically, the output of layer $l-1$ serves as the input to layer l, with transformations given by

$$Z_m^{(l)} = \sigma(w_{m0}^{(l)} + (w_m^{(l)})^\top Z^{l-1}),$$

for $m = 1, \dots, M_l$, where M_l denotes the number of units in the l-th layer. Note that each unit in the layer l is implicitly a function of the input X. Consequently, through successive non-linear transformations, the network constructs increasingly complex features of the original input X, which finally serve as inputs to the output layer.

4.2 Fitting neural networks

In this section, we discuss common ways of fitting the neural networks defined in the previous section. For notational convenience, we focus on the single-layer perceptron, but the methods extend naturally to the multi-layer perceptron, which we discuss briefly.

Consider the setup in Section 4.1. The parameters of the single-layer perceptron are referred to as weights, which we collect into a set θ. This set consists of

$$\begin{aligned} \{\alpha_{m0}, \alpha_m;\ m = 1, 2, \dots, M\} &\quad M(p+1) \text{ weights}, \\ \{\beta_{k0}, \beta_k;\ k = 1, 2, \dots, K\} &\quad K(M+1) \text{ weights}. \end{aligned}$$

Thus, the set θ contains $M(p+1) + K(M+1)$ parameters.

In both regression and classification problems, a single-layer perceptron neural network is fitted to training data, and the fit of the model is then assessed using an appropriate error function. We denote the set of training data by (x_i, y_i), $i = 1, \dots, N$, where $x_i = (x_{i1}, \dots, x_{ip})$, and $y_i = (y_{i1}, \dots, y_{iK})$. For regression problems, we seek to minimise the sum of squared errors:

$$R(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2. \tag{4.3}$$

The factor 1/2 often used in the literature is omitted here for simplicity. For classification problems, the fit of the model is evaluated using the negative multinomial log-likelihood, also referred to as the cross-entropy or log-loss, which is given by

$$R(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log\left(f_k(x_i)\right), \tag{4.4}$$

with the associated classifier $G(x) = \arg\max_k f_k(x)$. If the softmax function (4.2) is used in the output layer, the cross-entropy can be written as

$$R(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \left[\beta_{k0} + \beta_k^\top Z - \log\left(\sum_{\ell=1}^{K} \exp\left(\beta_{\ell 0} + \beta_\ell^\top Z\right)\right)\right],$$

which is a linear logistic regression model in the hidden units, the parameters of which can be estimated using maximum likelihood.

4.2.1 Backpropagation

The standard approach to minimise $R(\theta)$ is gradient descent, implemented by backpropagation (Hastie et al., 2009). Gradient descent is an iterative procedure that updates the parameters in the direction of steepest descent of the error function. Backpropagation exploits the compositional structure of the model to efficiently compute the partial derivatives using the chain rule, propagating gradients backward through the network from the output layer to the input layer.

We now present backpropagation for the squared error loss in (4.3) used in regression problems. A similar algorithm can be derived for K-class classification problems using the cross-entropy (4.4), since the computational components for cross-entropy share the same functional form as those of the sum of squared error function. Let $z_{mi} = \sigma(\alpha_{m0} + \alpha_m^\top x_i)$ denote the m-th hidden unit of observation i, and let $z_i = (z_{1i}, z_{2i}, \ldots, z_{Mi})^\top$ represent the full hidden layer of observation i. We can express the loss as

$$\begin{aligned} R(\theta) &\equiv \sum_{i=1}^{N} R_i \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2, \end{aligned}$$

and, by the chain rule, the partial derivatives of R_i with respect to the weights become

$$\begin{aligned} \frac{\partial R_i}{\partial \beta_{km}} &= \frac{\partial R_i}{\partial f_k(x_i)} \frac{\partial f_k(x_i)}{\partial T_{ki}} \frac{\partial T_{ki}}{\partial \beta_{km}} \\ &= -2(y_{ik} - f_k(x_i)) g_k'(\beta_{k0} + \beta_k^\top z_i) z_{mi}, \\ \frac{\partial R_i}{\partial \alpha_{m\ell}} &= \sum_{k=1}^{K} \frac{\partial R_i}{\partial f_k(x_i)} \frac{\partial f_k(x_i)}{\partial T_{ki}} \frac{\partial T_{ki}}{\partial z_{mi}} \frac{\partial z_{mi}}{\partial \alpha_{m\ell}}, \\ &= -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i)) g_k'(\beta_{k0} + \beta_k^\top z_i) \beta_{km} \sigma'(\alpha_{m0} + \alpha_m^\top x_i) x_{i\ell}, \end{aligned} \tag{4.5}$$

for $m = 1, \ldots, M$, and $\ell = 1, \ldots, p$. The above equations also hold for the bias terms α_{m0} and β_{k0}, i.e., $m = \ell = 0$, if we define $z_{0i} = x_{i0} = 1$. Note that this corresponds to adding an extra node in the input layer and a bias node feeding the output layer.

Using these gradients, the gradient descent update at iteration $(r+1)$ are given by

$$\begin{aligned} \beta_{km}^{(r+1)} &= \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}}, \\ \alpha_{m\ell}^{(r+1)} &= \alpha_{m\ell}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{m\ell}^{(r)}}, \end{aligned} \tag{4.6}$$

where γ_r denotes the learning rate. We can rewrite the gradients in (4.5) more compactly as

$$\begin{aligned} \frac{\partial R_i}{\partial \beta_{km}} &= \delta_{ki} z_{mi}, \\ \frac{\partial R_i}{\partial \alpha_{m\ell}} &= s_{mi} x_{i\ell}, \end{aligned} \tag{4.7}$$

where δ_{ki} and s_{mi} represent the 'errors' at the current iteration for the output layer and hidden layer, respectively. By definition, these error terms satisfy the backpropagation equations

$$s_{mi} = \sigma'(\alpha_{m0} + \alpha_m^\top x_i) \sum_{k=1}^{K} \beta_{km} \delta_{ki}, \tag{4.8}$$

which recursively propagate errors backward through the network. This leads to the following two-pass algorithm: In the forward pass, the current weights are fixed, and the fitted values $\hat{f}_k(x_i)$ are computed from (4.1). In the backward pass, the output errors δ_{ki} are computed and then backpropagated using (4.8) to obtain the hidden-layer errors s_{mi}. The error terms are then employed to compute the gradients in (4.7), which are subsequently used to update the weights according to (4.6). Note that all weights are updated simultaneously in each iteration.

As mentioned above, the algorithm naturally generalises to networks with multiple hidden layers by applying the backpropagation equations in (4.8) recursively. Specifically, the error terms are successively propagated backward through each hidden layer, where the error at a given layer depends on the weighted errors of the following layer and the derivative of the activation function. Assume, for simplicity, that each hidden layer contains the same number of hidden units M. Then, for each hidden layer l, the backpropagated error for hidden unit m in observation i is defined as

$$\delta_{mi}^{(l)} = \sigma'(\alpha_{m0}^{(l)} + (\alpha_m^{(l)})^\top z_i^{(l-1)}) \sum_{j=1}^{M} \delta_{ji}^{(l+1)} w_{jm}^{(l+1)},$$

where $w_{jm}^{(l+1)}$ denotes the weight connecting unit m in layer l to unit j in layer $l+1$. The corresponding partial derivative of R_i with respect to $w_{jm}^{(l+1)}$ is then

$$\frac{\partial R_i}{\partial w_{jm}^{(l+1)}} = \delta_{ji}^{(l+1)} z_{mi}^{(l)}.$$

4.2.2 Stochastic gradient descent

The gradient of the squared error loss function with respect to a single weight $w_{jm}^{(l)}$ is obtained by summing over all training observations and output units,

$$\frac{\partial R}{\partial w_{jm}^{(l)}} = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{\partial}{\partial w_{jm}^{(l)}} (y_{ik} - \hat{f}_k(x_i))^2.$$

Hence, computing a full gradient update requires evaluating all $N \times K$ terms for every weight in the network, which becomes computationally expensive as the number of parameters and observations increase. Furthermore, gradient descent typically requires many iterations to reach a local minimum (Hastie et al., 2009).

To alleviate these problems, the full gradient can be approximated by computing it from either a single observation or a small subset of the observations. This process is called stochastic gradient descent (SGD), since the resulting gradient estimate is inherently noisy and therefore only approximates the true gradient. In practice, observations are randomly shuffled and partitioned into mini-batches for training (Hastie et al., 2009). Each batch is then passed through the network, the loss is computed, the gradient of that loss is calculated, and the model weights are updated accordingly. After all batches have been processed once, one epoch has been completed. Increasing the number of epochs generally allows the model to fit the training data more closely. However, to mitigate overfitting, the number of epochs should be chosen with reference to validation performance based on the error on a validation dataset, rather than training performance alone.

Mini-batch SGD improves computational efficiency and thereby enables neural networks to scale to very large datasets. Moreover, the stochastic variability in the gradient estimates reduces the risk of the optimisation procedure becoming trapped in suboptimal local minima. This effect is reinforced by the practice of specifying the learning rate γ_r to decrease to zero, as $r \to \infty$, which permits relatively large updates in early iterations and then promotes stable convergence in later iterations. Convergence of the gradient is ensured if $\gamma_r \to 0$, $\sum_{r=1}^{\infty} \gamma_r = \infty$, and $\sum_{r=1}^{\infty} \gamma_r^2 < \infty$, (Hastie et al., 2009), which is, for example, satisfied when $\gamma_r = 1/r$.

4.3 Regularisation

When fitting the model, regularisation is generally applied either through a penalty term or by stopping the fitting process early, as a global minimum of $R(\theta)$ often results in overfitting. One way to account for this is to employ weight decay by introducing a penalty term $\lambda J(\theta)$ to the error function $R(\theta)$. The regularised error function is then given by

$$\tilde{R}(\theta) = R(\theta) + \lambda J(\theta),$$

where $\lambda \geq 0$ is a tuning parameter determined by cross-validation that controls the strength of regularisation.

Common choices for $J(\theta)$ include the L_1 (Lasso), L_2 (Ridge), and elastic net penalties. The L_1 penalty imposes an absolute value shrinkage on the weights,

$$J_{L_1}(\theta) = \sum_{k,m} |\beta_{km}| + \sum_{m,\ell} |\alpha_{m\ell}|,$$

while the L_2 penalty corresponds to the squared magnitude of the weights,

$$J_{L_2}(\theta) = \sum_{k,m} |\beta_{km}|^2 + \sum_{m,\ell} |\alpha_{m\ell}|^2.$$

The elastic net combines these two in a convex combination

$$J_{\mathrm{EN}}(\theta) = (1-\rho) J_{L_2}(\theta) + \rho J_{L_1}(\theta), \qquad \rho \in [0,1].$$

The three penalty types offer different shrinkage behaviour. The Lasso can perform variable selection by driving individual weights exactly to zero, thereby inducing sparsity in the model, but it is less stable in high-dimensional settings and performs poorly when predictors are strongly collinear. Ridge regularisation generally yields more stable optimisation, particularly under collinearity, but it does not perform variable selection. The elastic net combines the properties of both penalties by retaining the stabilising effect of Ridge while still allowing some degree of sparsity, and is therefore generally preferred when the predictor variables exhibit strong correlation. Figure A.2 illustrates the contours of the three regularisation penalties in a two-dimensional parameter space, where $R(\theta) = 1$, with the three unit balls corresponding to L_1, L_2, and elastic net regularisation, respectively.

In practice, it is often advantageous to specify a relatively large number of hidden units in each layer in order to provide sufficient representational capacity for capturing non-linear structure in the data. With appropriate regularisation, the influence of superfluous units can be reduced by shrinking their weights towards zero. Beyond explicit penalty-based regularisation, stochastic regularisation techniques such as dropout and dilution provide an alternative for controlling model complexity in neural networks. Dilution randomly shrinks individual weights towards zero, while dropout randomly sets the outputs of hidden units to zero. Both approaches can be interpreted as implicitly averaging over many randomly sparse network realisations. Like weight decay, these methods reduce overfitting by reducing sensitivity to individual parameters, but they operate through stochastic perturbations rather than through a deterministic penalty term.

While the preceding discussion follows the standard optimisation-based treatment of regularisation, we adopt a Bayesian perspective to motivate regularisation from a probabilistic standpoint rather than purely from an algorithmic one. In this framework, regularisation penalties arise from prior assumptions on the model parameters, allowing regularisation to be interpreted as a property of the underlying model assumptions rather than merely as a tuning mechanism.

Specifically, in a Bayesian formulation of a regression or prediction model, the parameters θ are treated as random variables with a prior distribution $p(\theta)$. Given data (X, Y) and assuming a likelihood $f(Y|X, \theta)$, Bayes' theorem yields the posterior distribution

$$p(\theta|X, Y) \propto f(Y|X, \theta)p(\theta),$$

where X is treated as fixed. Estimation is performed by selecting the posterior mode, also referred to as the maximum a posteriori (MAP) estimator,

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\theta|X, Y) = \arg\min_{\theta} \left\{-\log\left(f(Y|X, \theta)\right) - \log\left(p(\theta)\right)\right\}. \tag{4.9}$$

This shows the equivalence between MAP estimation and regularised empirical risk minimisation. The negative log-likelihood constitutes the data-fitting part, while the negative log-prior makes up the penalty term.

To illustrate this relation, consider a Gaussian prior with independent entries on the P parameters collected in θ,

$$\theta_j \sim \mathcal{N}(0, \tau^2), \qquad j = 1, \dots, P,$$

which implies

$$p(\theta) = (2\pi\tau^2)^{-P/2} \exp\left(-\frac{1}{2\tau^2}\sum_{j=1}^{P}\theta_j^2\right).$$

The negative log-prior is thus

$$-\log\left(p(\theta)\right) = \frac{P}{2}\log\left(2\pi\tau^2\right) + \frac{1}{2\tau^2}\sum_{j=1}^{P}\theta_j^2. \tag{4.10}$$

Since the additive constant in (4.10) does not affect the minimiser, the only term of relevance for optimisation is the second term. Substituting this into (4.9) yields

$$\hat{\theta}_{\mathrm{MAP}} = \arg\min_{\theta} \left\{ -\log\left(f(Y|X,\theta)\right) + \frac{1}{2\tau^2}\|\theta\|_2^2 \right\},$$

which coincides with an L_2-regularised optimisation problem,

$$\arg\min_{\theta}\{R(\theta) + \lambda\|\theta\|_2^2\},$$

with $\lambda = (2\tau^2)^{-1}$. The precision parameter $1/\tau^2$ controls the degree of shrinkage. A smaller prior variance produces stronger contraction of the weights towards zero, which is analogous to increasing the Ridge penalty.

A similar argument shows that the Lasso corresponds to a Laplace prior,

$$p(\theta_j) \propto \exp\left(-\frac{|\theta_j|}{b}\right),$$

for which the negative log-prior is proportional to $\|\theta\|_1$. Thus, the MAP estimator is given by

$$\hat{\theta}_{\mathrm{MAP}} = \arg\min_{\theta} \left\{ -\log\left(f(Y \mid X,\theta)\right) + \lambda\|\theta\|_1 \right\},$$

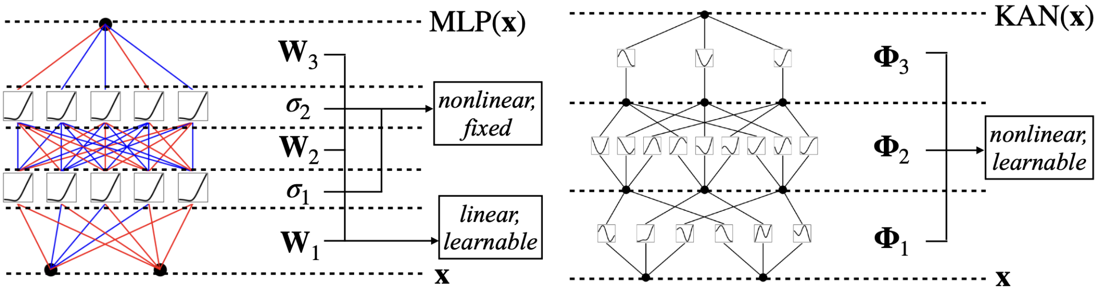
which corresponds to the Lasso penalty. Unlike the Gaussian prior, the Laplace prior is not conjugate to a Gaussian likelihood, meaning that the posterior does not reduce to a standard distribution with a closed-form representation. The posterior distribution therefore cannot be expressed analytically. Consequently, the posterior mode, that is the Lasso estimator, does not coincide with the posterior mean, which in contrast to the mode is generally not sparse.

4.4 Kolmogorov–Arnold Networks

Kolmogorov–Arnold Networks (KANs) are a recently proposed neural network architecture inspired by the Kolmogorov–Arnold representation theorem. In contrast to standard multi-layer perceptrons, where non-linearity is applied at the nodes of the network, KANs use learnable univariate functions as activation functions on the edges of the network graph. This yields an architecture where each connection implements a flexible non-linearity, while the nodes themselves perform only summation. Figure 4.2 provides a visual comparison of the architecture of KANs and MLPs.

The motivation behind KANs is to combine two complementary mechanisms for function approximation. Deep networks are effective at capturing compositional structure, that is, how variables interact through successive transformations. However, when using fixed activation functions such as ReLU, they may be inefficient at accurately approximating non-linear univariate components. Splines, by contrast, provide high-order and locally adaptive approximations of one-dimensional functions, but lack the structural flexibility required to efficiently represent high-dimensional compositional dependencies. KANs are designed to integrate these two aspects within a single framework. Empirical results reported by Liu et al. (2025) suggest that this design can improve predictive accuracy and interpretability relative to traditional MLPs, while also exhibiting more favourable neural scaling behaviour.

MLPs are based on the universal approximation theorem, which states that a sufficiently wide MLP can approximate any continuous function on a compact domain to arbitrary



(a) MLP (b) KAN

Figure 4.2: Comparison of multi-layer perceptron (MLP) and Kolmogorov–Arnold Network (KAN) architectures. Source: Liu et al. (2025).

accuracy. KANs are based on the Kolmogorov–Arnold representation theorem, which states that any continuous multivariate function can be represented as a finite superposition of univariate continuous functions and addition. For completeness, we state the Kolmogorov–Arnold representation theorem below.

Theorem 4.2 (Kolmogorov–Arnold representation theorem)
Let $n \geq 2$. There exist continuous univariate functions

$$\phi_{q,p} : [0,1] \to \mathbb{R}, \quad q = 1, \dots, 2n+1,\ p = 1, \dots, n,$$

such that any continuous multivariate function $f : [0,1]^n \to \mathbb{R}$ admits the representation

$$f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{4.11}$$

where each $\Phi_q : \mathbb{R} \to \mathbb{R}$ is continuous.

While mathematically elegant, the Kolmogorov–Arnold representation theorem is inherently non-constructive due to the pathological nature of the functions $\phi_{q,p}$ and Φ_q. Although the inner functions $\phi_{q,p}$ can be chosen independently of the target function f, depending only on the dimension n, they are highly non-smooth and may even exhibit fractal behaviour. In this sense, they resemble hashing mechanisms rather than well-behaved representations (Girosi and Poggio, 1989).

Moreover, since the outer functions Φ_q depend on the target function f, they can be at least as complex as f itself, and thus may not be approximated well by smooth parameterised representations. To address these limitations, Liu et al. (2025) propose extending the original two-layer representation in (4.11) to deeper architectures, thereby enabling more flexible and structured compositions of univariate functions and additive interactions.

4.4.1 Network architecture of KANs

This section defines the architecture of KANs based on Liu et al. (2025). For an intuitive visualisation of the architecture of KANs, we refer to Figure 4.2b. In the following, we consider a set of training data $\{(x_i, y_i)\}_{i=1}^{n}$ where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}^K$. The objective is to find a function $\hat{f}$ such that $\hat{f}(x_i) \approx y_i$ for all i.

The architecture of the KAN is specified by an integer array

$$[n_0, n_1, \ldots, n_L],$$

where n_ℓ denotes the number of nodes in the ℓ^{th} layer of the network. The first layer has $n_0 = p$ nodes corresponding to the input features, and the last layer has $n_L = K$ nodes corresponding to the output.

Denoting the i^{th} neuron in the ℓ^{th} layer by (ℓ, i), the activation function connecting node (ℓ, i) to node $(\ell+1, j)$ is denoted by

$$\phi_{\ell,j,i}, \qquad \ell = 0, \ldots, L-1, \qquad i = 1, \ldots, n_\ell, \quad j = 1, \ldots, n_{\ell+1},$$

which is a learnable univariate function parametrised as a B-spline curve. There are $n_\ell n_{\ell+1}$ activation functions between layer ℓ and $\ell+1$. The pre-activation of $\phi_{\ell,j,i}$ is denoted by $x_{\ell,i}$ and the post-activation is $\tilde{x}_{\ell,j,i} \coloneqq \phi_{\ell,j,i}(x_{\ell,i})$. The activation at node $(\ell+1, j)$ is then the sum of incoming post-activations, given by

$$x_{\ell+1,j} = \sum_{i=1}^{n_\ell} \tilde{x}_{\ell,j,i} = \sum_{i=1}^{n_\ell} \phi_{\ell,j,i}(x_{\ell,i}), \tag{4.12}$$

for $j = 1, \ldots, n_{\ell+1}$ and $\ell = 0, \ldots, L-1$. Figure 4.3 illustrates the activations in a KAN layer. Note that the activation at each node is a sum of univariate functions applied to the activations of the previous layer, and there are no learnable parameters at the nodes themselves. The learnable parameters are contained entirely within the univariate functions on the edges.

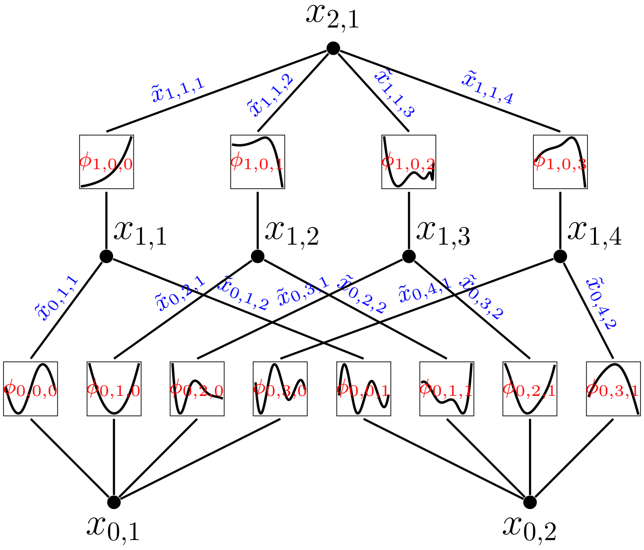


Figure 4.3: Activations in a KAN layer. Each edge from node (ℓ, i) to node $(\ell+1, j)$ is associated with a learnable univariate function $\phi_{\ell,j,i}$ that transforms the pre-activation $x_{\ell,i}$ into the post-activation $\tilde{x}_{\ell,j,i}$. The activation at node $(\ell+1, j)$ is the sum of incoming post-activations.

In matrix form, (4.12) can be expressed as

$$x_{\ell+1} = \Phi_\ell(x_\ell) = \begin{pmatrix} \phi_{\ell,1,1} & \phi_{\ell,1,2} & \cdots & \phi_{\ell,1,n_\ell} \\ \phi_{\ell,2,1} & \phi_{\ell,2,2} & \cdots & \phi_{\ell,2,n_\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{\ell,n_{\ell+1},1} & \phi_{\ell,n_{\ell+1},2} & \cdots & \phi_{\ell,n_{\ell+1},n_\ell} \end{pmatrix} x_\ell,$$

where Φ_ℓ denotes the matrix of univariate functions acting componentwise in the ℓ^{th} layer in (4.12).

A KAN with L layers thus admits the composition

$$\mathrm{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0)(x). \tag{4.13}$$

Expanding (4.13) yields the following explicit expression for the output of the KAN:

$$\mathrm{KAN}(x) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_L,i_{L-1}} \Bigg(\sum_{i_{L-2}=1}^{n_{L-2}} \cdots \Bigg(\sum_{i_2=1}^{n_2} \phi_{2,i_3,i_2} \Bigg(\sum_{i_1=1}^{n_1} \phi_{1,i_2,i_1} \Bigg(\sum_{i_0=1}^{n_0} \phi_{0,i_1,i_0}(x_{i_0}) \Bigg)\Bigg)\Bigg) \cdots \Bigg),$$

which is a superposition of univariate functions and addition. As such, the original Kolmogorov–Arnold representation in (4.11) corresponds to a two-layer KAN with architecture $[n, 2n+1, 1]$. Since all the operations employed in KANs are differentiable, the network can be trained using backpropagation and gradient-based optimisation algorithms.

Comparing the composition (4.13) with that of MLPs described in Section 4.1, the output of a depth-L MLP can be written as

$$\mathrm{MLP}(x) = (W_{L-1} \circ \sigma \circ W_{L-2} \circ \sigma \circ \cdots \circ W_0)(x),$$

where W_ℓ are affine maps and σ is a fixed activation function. The difference is that in MLPs, the linear transformations W_ℓ and the non-linear activation σ are separate operations, while in KANs, the linear and non-linear transformations are unified in the function matrices Φ_ℓ. In MLPs, the non-linearity is fixed and applied at the nodes, while in KANs, the non-linearity is learnable and applied on the edges.

Example 4.3
Suppose we want to approximate the function

$$f(x_1, x_2) = x_1 x_2$$

using a KAN. We need to represent f as a composition of univariate functions and addition. Using the identity

$$x_1 x_2 = \frac{1}{4}\left((x_1+x_2)^2 - (x_1-x_2)^2\right), \tag{4.14}$$

we can express f by choosing an $L = 2$ layer KAN with architecture $[2, 2, 1]$ as displayed in Figure 4.4. The first layer computes the sums and differences of the inputs, that is

$$\phi_{0,1,1}(x) = \phi_{0,1,2}(x) = x, \qquad \phi_{0,2,1}(x) = x, \quad \phi_{0,2,2}(x) = -x,$$

while the second layer applies the squaring function to these intermediate activations,

$$\phi_{1,1,1}(x) = \frac{1}{4}x^2, \qquad \phi_{1,1,2}(x) = -\frac{1}{4}x^2.$$

The final layer then sums these activations to yield the output $f(x_1, x_2)$.

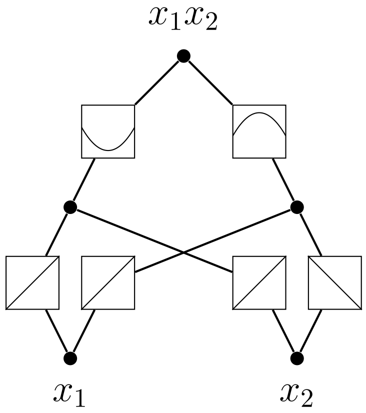


Figure 4.4: Example of a KAN representation of the function $f(x_1, x_2) = x_1x_2$ using the identity in (4.14). The first layer computes the sums and differences of the inputs, while the second layer applies the squaring function to these intermediate activations. The final layer sums these activations to yield the output $f(x_1, x_2)$.

The representation in (4.14) of f is not unique. For instance, we could also represent f using exponentials and logarithms as follows:

$$x_1x_2 = \exp\left(\log\left(x_1\right) + \log\left(x_2\right)\right), \tag{4.15}$$

which corresponds to a KAN with architecture $[2, 1, 1]$ where the first layer applies the logarithm function to each input, and the second layer applies the exponential function to the sum of the intermediate activations. The representation using (4.15) contains one less node in the hidden layer compared to the one in (4.14) at the cost of using more complex univariate functions. In particular, the activations may be more difficult to learn due to the non-linearity of the logarithm and exponential functions, especially if the inputs can take values close to zero or negative values. This illustrates the trade-off between depth and complexity of univariate functions in KANs, where deeper architectures can use simpler univariate functions, while shallower architectures may require more complex univariate functions to capture the same target function.

4.4.2 Parametrisation of activation functions

Example 4.3 provides a simple example of how a KAN can represent a function as a composition of univariate functions and addition. However, as noted in the discussion following Theorem 4.2, the resulting univariate functions need not be well-behaved and may be difficult to learn. To address this issue, KANs employ a residual parametrisation of activation functions in which each univariate function ϕ is represented as a spline expansion augmented by a smooth baseline activation. Specifically,

$$\phi(x) = w_b b(x) + w_s \sum_{i=1}^{G} c_i B_i(x),$$

where $b(x)$ denotes a fixed baseline activation function, chosen here as the SiLU function,

$$b(x) = \frac{x}{1 + e^{-x}},$$

B_i are B-spline basis functions defined on a grid of size G which may be adapted during training, and c_i, w_b, and w_s are trainable parameters. The scaling parameters w_b and w_s are included to allow amplitude control of the components, even though these could be absorbed into $b(x)$ and the spline component. The spline grid is updated during training to ensure that the spline component can adapt to the input activations, and the B-spline basis functions are defined on this grid.

This residual parametrisation provides a stable non-linear reference at initialisation, since the SiLU activation already exhibits smooth non-linearity and well-behaved derivatives. Consequently, the spline component primarily learns a residual correction rather than the full activation function from scratch. This can improve optimisation stability and reduce sensitivity to parameter initialisation compared to a purely spline-based parametrisation without a baseline.

Assume a KAN architecture with depth L and equal width $n_0 = n_1 = \cdots = n_L = N$, and suppose that each edge activation is parametrised by a spline of order k defined on G intervals. Between two consecutive layers there are N^2 edges, so the total number of edge activations scales as $N^2 L$. Since each spline requires $G + k$ parameters, the total number of trainable parameters is

$$O(N^2 L(G + k)).$$

As the spline order k is usually chosen as $k = 3$ to balance approximation power, computational efficiency, and optimisation stability, the overall parameter count scales as

$$O(N^2 LG).$$

In contrast, a standard MLP of the same depth and width requires $O(N^2 L)$ parameters, reflecting the additional flexibility introduced by the spline parametrisation in KANs. However, as noted in (Liu et al., 2025), KANs usually require fewer observations to achieve the same level of accuracy compared to MLPs, which can mitigate the increased parameter count.

4.4.3 Approximation theory of KANs

In this section, we discuss the scaling behaviour of KANs based on the approximation theory of KANs. The following theorem establishes that if a function admits a representation as a composition of sufficiently smooth univariate functions arranged in KAN architecture, then it can be approximated by a KAN with spline activations, with an approximation error that decreases at a rate determined by the grid size G and spline order k.

Theorem 4.4 (Approximation theory, KAT)
Let $x = (x_1, x_2, \cdots, x_n)$, and suppose that a function $f(x)$ admits a representation

$$f(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)x,$$

as in (4.13), where each function $\Phi_{\ell,j,i}$ is $(k+1)$-times continuously differentiable. Then there exists a constant C, such that for any grid size G, there exist k-th order B-spline functions $\Phi^G_{\ell,j,i}$ such that for any $0 \le m \le k$, the approximation error of f by the KAN with spline activations measured in the C^m-norm is bounded by

$$\|f(x) - (\Phi^G_{L-1} \circ \Phi^G_{L-2} \circ \cdots \circ \Phi^G_1 \circ \Phi^G_0)x\|_{C^m} \le CG^{-k-1+m}, \tag{4.16}$$

where the C^m-norm measures the magnitude of derivatives up to order m:

$$\|g\|_{C^m} = \max_{|\beta| \le m} \sup_{x \in [0,1]^n} \left| D^\beta g(x) \right|.$$

Remark: It should be emphasised that the functions $\phi_{q,p}$ and Φ_q in the Kolmogorov–Arnold representation (4.11) need not be smooth and may be highly irregular. Consequently, the approximation theory for KANs based on smooth spline activations does not apply directly to Theorem 4.2. The theorem does, however, allow for arbitrary depth and width. In practice, the performance of KANs therefore depends on whether the target function admits a sufficiently smooth compositional representation. Liu et al. (2025) suggest increasing the depth and width of the network, which may enable the model to uncover smoother intermediate representations of the target function.

Proof: By one-dimensional B-spline approximation theory (de Boor, 1978, Ch. 12), for each $\Phi_{\ell,j,i}$ there exists a spline function $\Phi^G_{\ell,j,i}$ of order k on a grid with G knots such that for every $0 \le m \le k$,

$$\|\Phi_{\ell,j,i} - \Phi^G_{\ell,j,i}\|_{C^m} \le C_{\ell,j,i}\, G^{-(k+1-m)}, \tag{4.17}$$

where the constant $C_{\ell,j,i}$ depends on derivatives of $\Phi_{\ell,j,i}$ up to order $k+1$ but is independent of G. We need to show that this approximation of the spline functions is bounded in the C^m-norm when composed with the other layers of the KAN.

Fix ℓ and denote by Φ^G_ℓ the layer obtained by replacing each $\Phi_{\ell,j,i}$ with $\Phi^G_{\ell,j,i}$. Define

$$u(x) := (\Phi_{\ell-1} \circ \cdots \circ \Phi_0)(x).$$

Then

$$(\Phi_\ell \circ u)(x) - (\Phi^G_\ell \circ u)(x) = (\Phi_\ell - \Phi^G_\ell)(u(x)). \tag{4.18}$$

By the multivariate chain rule, derivatives of (4.18) of order at most m can be expressed as finite sums of products involving derivatives of u and derivatives of $\Phi_\ell - \Phi^G_\ell$. Since the input domain is compact and each layer map is continuous, the image of the domain under any finite composition of layers is compact. Moreover, because each layer map is C^{k+1}, the composition u is C^{k+1}, and its derivatives up to order m are therefore uniformly bounded. Hence, there exists a constant C independent of G such that

$$\|(\Phi_\ell - \Phi^G_\ell) \circ u\|_{C^m} \le C\, \|\Phi_\ell - \Phi^G_\ell\|_{C^m}.$$

Combining this with the spline approximation bound in (4.17) yields

$$\|(\Phi_\ell \circ u) - (\Phi^G_\ell \circ u)\|_{C^m} \le C\, G^{-(k+1-m)}.$$

Define the residual of replacing Φ_ℓ with Φ^G_ℓ as

$$\begin{aligned} R_\ell(x) := {} & (\Phi^G_{L-1} \circ \cdots \circ \Phi^G_{\ell+1} \circ \Phi_\ell \circ \Phi_{\ell-1} \circ \cdots \circ \Phi_0)(x) \\ & - (\Phi^G_{L-1} \circ \cdots \circ \Phi^G_{\ell+1} \circ \Phi^G_\ell \circ \Phi_{\ell-1} \circ \cdots \circ \Phi_0)(x) \\ = {} & (\Phi^G_{L-1} \circ \cdots \circ \Phi^G_{\ell+1}) \circ ((\Phi_\ell - \Phi^G_\ell) \circ u)(x). \end{aligned}$$

By the triangle inequality,

$$\|\Phi_\ell^G\|_{C^m} \le \|\Phi_\ell\|_{C^m} + \|\Phi_\ell - \Phi_\ell^G\|_{C^m}.$$

From (4.17),

$$\|\Phi_\ell - \Phi_\ell^G\|_{C^m} \le C_\ell G^{-(k+1-m)} \le C_\ell,$$

for $G \ge 1$, and therefore,

$$\|\Phi_\ell^G\|_{C^m} \le \|\Phi_\ell\|_{C^m} + C_\ell,$$

which shows that $\|\Phi_\ell^G\|_{C^m}$ is uniformly bounded in G. Consequently, the derivatives of $\Phi_{\ell+1}^G, \dots, \Phi_{L-1}^G$ up to order m are uniformly bounded in G on compact sets. By repeated application of the multivariate chain rule, composition with a map whose derivatives up to order m are uniformly bounded defines a bounded operator on C^m. Therefore, there exists a constant C, independent of G, such that

$$\|R_\ell\|_{C^m} \le C\, G^{-(k+1-m)}.$$

Defining

$$F^{(\ell)}(x) := (\Phi_{L-1}^G \circ \cdots \circ \Phi_{\ell+1}^G \circ \Phi_\ell \circ \cdots \circ \Phi_0)(x)$$

with the convention that $F^{(-1)}(x) = (\Phi_{L-1}^G \circ \cdots \circ \Phi_0^G)(x)$, it follows that

$$R_\ell(x) = F^{(\ell)}(x) - F^{(\ell-1)}(x),$$

and thus, we observe that successive layer replacement yields a telescoping sum of the form

$$f(x) - (\Phi_{L-1}^G \circ \cdots \circ \Phi_1^G \circ \Phi_0^G)(x) = \sum_{\ell=0}^{L-1} R_\ell(x).$$

Taking the C^m-norm and applying the triangle inequality gives

$$\|f - (\Phi_{L-1}^G \circ \cdots \circ \Phi_1^G \circ \Phi_0^G)\|_{C^m} \le \sum_{\ell=0}^{L-1} \|R_\ell\|_{C^m} \le LC\, G^{-(k+1-m)}.$$

Absorbing L into the constant yields

$$\|f - (\Phi_{L-1}^G \circ \cdots \circ \Phi_1^G \circ \Phi_0^G)\|_{C^m} \le C' G^{-(k+1-m)},$$

which establishes the desired approximation bound (4.16). ■

The approximation result in Theorem 4.4 can be related to empirical findings on neural scaling laws, where the test error R, defined as the root mean squared error (RMSE), often follows a power law in the number of parameters N, i.e., $R \propto N^{-\alpha}$. Results from approximation theory show that if one approximates functions on an input manifold of intrinsic dimension d by piecewise polynomials of order k, then $\alpha = (k+1)/d$, reflecting the curse of dimensionality (Sharma and Kaplan, 2020).

In contrast, the bound in (4.16) yields an approximation rate $G^{-(k+1-m)}$ that is independent of the input dimension n, since the compositional structure of KANs relies solely on spline approximation of univariate functions. This suggests that, under the structural assumptions of Theorem 4.4, KANs do not suffer from the curse of dimensionality at the level of approximation rates. By comparison, approximation results for MLPs show that in some cases the approximation rate deteriorates exponentially with n (Lin et al., 2017). For fixed depth and width, the number of parameters grows linearly in the grid size

G, implying an effective scaling exponent $\alpha = k + 1$. It should be noted, however, that the constant C in (4.16) depends on the specific representation of f and may therefore still depend on n.

This interpretation highlights a notable distinction from the universal approximation theorem underlying MLPs. While the UAT guarantees that any continuous function can be approximated arbitrarily well by an MLP, it does not quantify how $N(\varepsilon)$ scales with the approximation error, and worst-case approximation rates over general smoothness classes may still deteriorate with dimension. In contrast, Theorem 4.4 provides a conditional result: if the target function admits a sufficiently smooth compositional structure, then the approximation rate is governed by the univariate spline resolution rather than the ambient input dimension.

Empirically, the performance of KANs typically depends on whether the target function admits a sufficiently smooth compositional representation. Liu et al. (2025) demonstrate the accuracy and efficiency of KANs across a wide range of functions in their empirical studies. They generally observe that the test RMSE scales as G^{-4}, which is consistent with Theorem 4.4, given that the degree of the spline is chosen as $k = 3$. In particular, they report that KANs represent several special functions more efficiently and accurately than MLPs (Liu et al., 2025, Table 1).

However, the observed scaling of the test RMSE is in some cases closer to G^{-3} or even slower. They attribute this to optimisation error and to larger approximation errors when evaluating the model near the boundary of the input domain (Michaud et al., 2023). Indeed, when evaluating the root median squared error instead of the RMSE, they observe a scaling closer to G^{-4}, which suggests that the slower scaling of the RMSE may be driven by outliers in the test set.

4.4.4 Grid Extension

The implementation of KANs described in Liu et al. (2025) utilises grid extension during training to improve the approximation accuracy. The idea is to increase accuracy of the spline approximation by refining the grid, while preserving the function represented by the previously trained coarse spline. This allows the model capacity to be increased without retraining the spline model from scratch.

Consider a univariate function $f : [a, b] \to \mathbb{R}$ that we approximate by B-splines of order k. Let the initial coarse grid consist of G_1 intervals with knots $t_0 = a < t_1 < \cdots < t_{G_1} = b$, augmented with k boundary knots to the grid $\{t_{-k}, \ldots, t_{G_1+k}\}$. The corresponding spline space has dimension $G_1 + k$, with basis functions $B_i, i = 0, \ldots, G_1 + k - 1$, such that we approximate f by a coarse spline of the form $f_{\text{coarse}}(x) = \sum_{i=0}^{G_1+k-1} c_i B_i(x)$, where $c_i \in \mathbb{R}$ are the learned coefficients.

The grid is then refined to a finer grid with $G_2 > G_1$ intervals, producing a new B-spline basis $B'_j, j = 0, \ldots, G_2 + k - 1$ such that the spline on the refined grid takes the form $f_{\text{fine}}(x) = \sum_{j=0}^{G_2+k-1} c'_j B'_j(x)$, with new coefficients c'_j. Initially, the coefficients c'_j are chosen such that f_{fine} approximates f_{coarse} as closely as possible, which can be achieved by solving the least-squares projection problem

$$\arg\min_{\{c'_j\}} \mathbb{E}_{x \sim p(x)} \left(\sum_{j=0}^{G_2+k-1} c'_j B'_j(x) - \sum_{i=0}^{G_1+k-1} c_i B_i(x) \right)^2,$$

where $x \sim p(x)$ denotes that x is sampled from the empirical distribution of the spline pre-activations, obtained by propagating the training data through the current network.

Since the spline space associated with the fine grid contains the coarse spline space, the projection typically reproduces the coarse function up to numerical error. After this reparameterisation, training can continue on the refined grid. In this way, grid extension allows increased model complexity while retaining the structure learned during earlier optimisation stages.

4.4.5 Regularisation and pruning of KANs

Since the number of parameters in KANs scales as $O(N^2LG)$ compared to $O(N^2L)$ for MLPs (see Subsection 4.4.2), regularisation and pruning techniques are important to reduce overfitting and improve generalisation.

Similar to L_1-regularisation described in Section 4.3, a penalty term can be added to the loss function to encourage sparsity in the model. Since KANs contain activation functions rather than linear weights, we define the L_1-norm of any activation function as the average magnitude over its N_p evaluation points $x^{(i)}$:

$$|\phi|_1 := \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \phi(x^{(i)}) \right| .$$

The L_1-norm of a layer map Φ with n_{in} input nodes and n_{out} output nodes is then defined as

$$|\Phi|_1 := \sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} |\phi_{i,j}|_1 ,$$

and the entropy of Φ is defined as

$$S(\Phi) := - \sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} \frac{|\phi_{i,j}|_1}{|\Phi|_1} \log \left(\frac{|\phi_{i,j}|_1}{|\Phi|_1} \right) .$$

A higher entropy indicates that the activation magnitudes are distributed more uniformly across edges, whereas a lower entropy indicates that the activity of the layer is concentrated on a smaller subset of activation functions. Encouraging lower entropy therefore promotes sparse representations, which can improve both interpretability and generalisation.

Using the L_1- and entropy-based regularisation, the regularised loss function for training KANs can be written as the sum of the original prediction loss R_{pred} and the regularisation terms:

$$\tilde{R} = R_{\text{pred}} + \lambda \left(\mu_1 \sum_{\ell=0}^{L-1} |\Phi_\ell|_1 + \mu_2 \sum_{\ell=0}^{L-1} S(\Phi_\ell) \right) \tag{4.19}$$

where $\lambda \geq 0$ is a tuning parameter typically selected via cross-validation, and $\mu_1, \mu_2 \geq 0$ are hyperparameters that balance the contributions of the L_1-norm and entropy regularisation terms, though usually set to $\mu_1 = \mu_2 = 1$.

The L_1-norm penalty encourages the overall magnitude of the activation functions to be small, while the entropy regularisation encourages the remaining magnitude to be concentrated on a small number of edges rather than distributed evenly across many. Together, these terms promote sparse and structured representations within each layer.

In addition to regularisation, which trains the model using the penalised loss in (4.19), pruning techniques can be applied to further reduce model complexity. This can improve generalisation by removing redundant components of the network and encouraging simpler functional representations.

Interestingly, larger architectures do not necessarily achieve better test performance, even though they are strictly more expressive. For example, Liu et al. (2025, Figure 2.3) report that a $[2, 1, 1]$ KAN achieves a lower test RMSE than a $[2, 5, 1]$ KAN on the synthetic task of approximating $f(x_1, x_2) = \exp\left(\sin \pi x_1 + x_2^2\right)$, which can be represented exactly by a $[2, 1, 1]$ KAN. Although the larger architecture contains the smaller one as a special case and is therefore capable of representing the same function, the additional parameters can make optimisation more difficult and may lead the model to learn unnecessarily complex representations.

In practice, however, the minimal KAN architecture is rarely known in advance. A common strategy is therefore to train a larger model and subsequently prune unnecessary components to obtain a simpler effective architecture. To this end, Liu et al. (2025) propose pruning nodes rather than edges. In particular, neuron i in layer ℓ is assigned an input and output score defined as

$$I_{\ell,i} = \max_k \left(|\phi_{\ell-1,i,k}|_1\right), \qquad O_{\ell,i} = \max_j \left(|\phi_{\ell,j,i}|_1\right),$$

and if either score is below a pruning threshold $\tau > 0$, then neuron i in layer ℓ is pruned by setting all its incoming and outgoing activations to zero. This effectively removes the neuron from the network, reducing the model complexity while retaining the remaining structure. By default, the pruning threshold is set to $\tau = 0.01$. After pruning, the model can be fine-tuned on the remaining parameters to recover any lost performance, for instance by grid extension.

5 | Forward curve construction and spot price calibration

In this chapter, we address two related objectives. First, we construct a daily forward curve from the monthly and quarterly contracts observed in the market. Second, we calibrate the spot price model developed in Chapter 2 to this curve so that risk-neutral expected future spot prices are consistent with observed forward prices at all maturities. Together, these steps provide the market-consistent price inputs required for the valuation methods developed in Chapter 6.

Because gas forwards are quoted for delivery periods rather than individual days, they cannot be used directly in storage valuation, where operational decisions are made daily. Assigning a single price to each contract period would create artificial jumps at contract boundaries and ignore the intra-period variation implied by seasonality. We therefore begin with the simple fixed spread approach in Subsection 5.1.1, which introduces weekday–weekend variation while preserving the quoted contract prices, and then turn to the spline-based smoothing procedure of Benth et al. (2007) in Section 5.1, which yields a globally smooth curve shaped by the estimated seasonal component.

Once the forward curve has been constructed, the spot price simulation is calibrated so that the expected spot price under the risk-neutral measure $\mathbb{Q}$ equals the observed forward price at each maturity. This calibration step follows the decomposition of the log-price into a deterministic risk-neutral mean path and a zero-mean Ornstein–Uhlenbeck deviation process, as derived in Section 2.1. The resulting simulated paths are therefore consistent with the forward curve and can be used directly in the spot-based valuation methods of Chapter 6.

5.1 Daily forward curve construction

This section develops the daily forward curve used throughout the valuation analysis. We first present a simple fixed spread approach that introduces weekday–weekend variation, and then the spline-based method of Benth et al. (2007), which decomposes the curve into a seasonal component and a smooth correction term chosen to satisfy the observed contract prices. As established in Chapter 2, the seasonal component is specified parametrically from historical spot data, while the correction term is obtained by solving a constrained optimisation problem that minimises the integrated squared second derivative subject to the contract consistency constraints. As shown in Theorem 5.1, the solution is a quartic polynomial spline.

The resulting forward curve serves two purposes in this thesis. It provides the deterministic price inputs for the intrinsic and rolling intrinsic valuation approaches of Sections 6.2 and 6.3, and it serves as the calibration target for the spot price simulations used in the LSMC-based valuation methods of Section 3.3. The methods are applied to $m = 16$ observed contract closing prices, consisting of 12 monthly and 4 quarterly contracts for delivery between 2024 and 2025. As stated in Subsection 1.2.1, using closing prices may overestimate intrinsic value. However, since bid–ask spreads are typically in the range of 1–2% of the prevailing price, they have only a limited effect on the valuation results presented in our examples (Breslin et al., 2008). In periods of reduced market liquidity, these spreads may widen and have a more material impact, in which case using mid-market prices or incorporating bid–ask spreads into the valuation framework may be necessary.

For a discussion on incorporating bid–ask spreads into the spline-based approach for construction of forward curves, we refer the reader to Benth et al. (2007).

5.1.1 Fixed spread approach

We begin with a simple forward curve construction that imposes a constant spread between weekday and weekend delivery prices. The approach is computationally straightforward and captures the main intraweek seasonality in gas markets. Under this assumption, the quoted contract price can be decomposed into piecewise-constant weekday and weekend levels.

Let F_i^C denote the observed price of contract i with delivery interval $[T_i^s, T_i^e]$, and let N_i be the number of settlement days in that interval. Partition the settlement days into weekday and weekend sets, with cardinalities $n_{wd,i}$ and $n_{we,i}$ respectively, so that $N_i = n_{wd,i} + n_{we,i}$. Under end-of-period settlement, the contract price is

$$F_i^C = \frac{n_{wd,i}\, f_{wd,i} + n_{we,i}\, f_{we,i}}{N_i},$$

where $f_{wd,i}$ and $f_{we,i}$ are the (piecewise-constant) weekday and weekend forward levels over contract i respectively. The fixed spread assumption is

$$f_{wd,i} - f_{we,i} = s,$$

with constant spread s estimated from market data. Solving the two equations gives

$$f_{we,i} = F_i^C - \frac{n_{wd,i}}{N_i} s, \qquad f_{wd,i} = F_i^C + \frac{n_{we,i}}{N_i} s.$$

This approach provides a practical middle ground between a fully continuous curve and a simple piecewise-constant approach. It incorporates intraweek seasonality while avoiding the complexity of constructing a fully continuous curve.

Figure 5.1 presents the resulting forward curve under the fixed spread assumption. Compared to the spline-based approach, the fixed spread method produces a simpler price structure but introduces artificial jumps at contract boundaries. The spline-based method, on the other hand, yields a globally smooth curve that captures the seasonal pattern while ensuring consistency with observed contract prices.

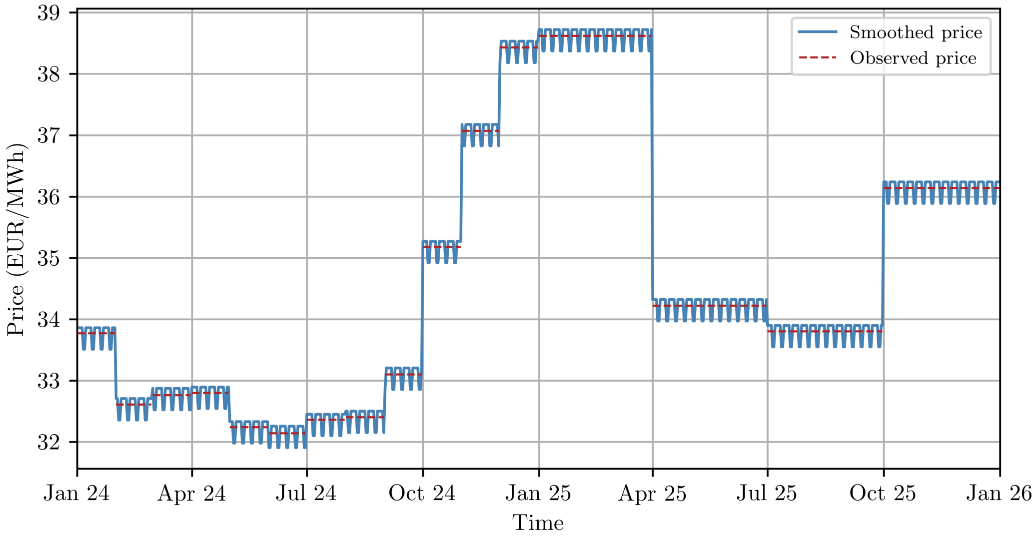


Figure 5.1: Forward curve constructed using a fixed spread approach, with separate weekday and weekend price levels. The constant spread captures the systematic intraweek seasonality in market contracts.

5.1.2 Maximum-smoothness spline approach

In this section, we follow the approach outlined in Benth et al. (2007) for constructing a smooth daily forward curve from observed futures prices. The methodology is to decompose the curve into a seasonal component and a residual (or correction) term. The seasonal component is specified a priori by a parametric function, while the correction term is constructed based on a maximum smoothness condition. As we will show, this turns out to be a quartic polynomial spline.

The algorithm is formulated for average-based forward contracts. For such contracts, the settlement price is defined as the arithmetic average of the underlying daily spot or day-ahead gas prices over the full delivery period, rather than a price fixed at a single delivery date. Additionally, delivery occurs at a constant rate throughout the settlement period.

Throughout, we work under the risk-neutral measure $\mathbb{Q}$, as motivated in Section 2.1. Consider a forward contract with delivery period from T^s to T^e, with an observed market price of $F(\tau, T^s, T^e)$ at time τ for receiving one unit of gas at a constant rate over the delivery period. The time-τ-value of the contract is given by

$$\int_{T^s}^{T^e} e^{-r(u-\tau)}[S(u) - F(\tau, T^s, T^e)]\,\mathrm{d}u,$$

where $S(u)$ denotes the spot price at time u and r is the risk-free interest rate. Since the forward contract is entered into at zero cost, its value at time τ must satisfy the arbitrage-free condition

$$\mathbb{E}_{\mathbb{Q}}\left[\int_{T^s}^{T^e} e^{-r(u-\tau)}\left[S(u) - F(\tau, T^s, T^e)\right]\,\mathrm{d}u \middle| \mathcal{F}_\tau\right] = 0. \tag{5.1}$$

It is well-known that the price at time τ of a fixed-delivery forward with delivery at the fixed time $t \geq \tau$ is given by

$$f(\tau, t) = \mathbb{E}\left[S(t)|\mathcal{F}_\tau\right]. \tag{5.2}$$

Using (5.2) and assuming that the contract is marked-to-market, which means that settlement occurs continuously over the delivery period, (5.1) can be rearranged to express the forward price as

$$F(\tau, T^s, T^e) = \int_{T^s}^{T^e} w(r;t) f(\tau, t)\,\mathrm{d}t, \tag{5.3}$$

where the weighting function $w(r;t)$ is defined as

$$w(r;t) = \frac{e^{-r(t-\tau)}}{\int_{T^s}^{T^e} e^{-r(u-\tau)}\,\mathrm{d}u}.$$

If instead, the contract is settled at the end of the delivery period, the weighting function simplifies to $w = 1/(T^e - T^s)$. The representation in (5.3) shows that the forward price is a discount-weighted average of expected future spot prices.

In practice, settlement does not occur continuously over the delivery period, but rather at a finite number of discrete times. Suppose that settlement takes place at N time points, $t_1 < t_2 < \cdots < t_N$, where $t_1 = T^s$ and $t_N = T^e$. In this case, the arbitrage-free price of the average-based forward contract can be written as

$$F(\tau, T^s, T^e) = \sum_{k=1}^{N} w(r;t_k)\, f(\tau, t_k),$$

where $f(\tau, t_k) = \mathbb{E}[S(t_k) \mid \mathcal{F}_\tau]$ denotes the price at time τ of a fixed-delivery forward with delivery at time t_k, and the discrete-time weighting function is given by

$$w(r;t_k) = \frac{e^{-r(t_k-\tau)}}{\sum_{j=1}^{N} e^{-r(t_j-\tau)}}. \tag{5.4}$$

Note from the above expressions that average-based forward contracts can be interpreted as swap contracts, exchanging a fixed price for a floating spot price over the delivery period. The floating leg consists of the realized spot prices during delivery, while the fixed leg consists of payments at the constant rate $F(\tau, T^s, T^e)$. If instead the contract is settled only once at the terminal delivery time T^e, the weighting function simplifies to $w(r;t_k) = \frac{1}{N}, k = 1, \ldots, N$.

In the following, we assume $\tau = 0$ and drop the dependence on τ in the notation, writing

$$F(T^s, T^e) = \sum_{k=1}^{N} w(r;t_k) f(t_k), \tag{5.5}$$

for discrete delivery. Suppose we observe m closing market prices $F_i^C := F(T_i^s, T_i^e)$ for average-based forward contracts with delivery periods $[T_i^s, T_i^e]$, $i = 1, \ldots, m$, where the notation C indicates that these are closing market prices. In case the delivery periods overlap, we construct a common settlement grid by collecting all unique delivery dates $\{t_0, t_1, \ldots, t_n\}$ implied by the contracts, where the grid $\{t_0, t_1, \ldots, t_n\}$ contains $n+1$ distinct dates across the m contracts. This is illustrated in Figure 5.2.

Each contract price can then be written in the form (5.5), with weights determined by its delivery interval and zero outside the corresponding interval.

We now seek to specify a functional form for the forward prices $f(t_k)$ that is consistent with the m observed contract prices in (5.5). To this end, we decompose the forward curve into a seasonal component $s(t)$, representing systematic seasonal variation, and a correction term $\varepsilon(t)$ capturing deviations from this seasonal pattern:

$$f(t) = s(t) + \varepsilon(t), \quad t \in [t_0, t_n]. \tag{5.6}$$

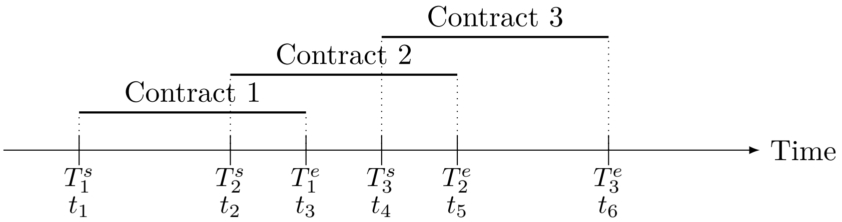


Figure 5.2: Construction of the common settlement grid from the delivery periods of observed average-based forward contracts. The grid consists of all unique delivery dates across the contracts, which are used as the time points for evaluating the forward curve.

While the forward price $F(\tau, T^s, T^e)$ is defined under the risk-neutral measure $\mathbb{Q}$, in many commodity markets, such as the gas market, there is no unique arbitrage-free relationship between spot price dynamics and the forward curve due to market incompleteness and limited storability (Benth et al., 2007), so the risk-neutral pricing measure is not uniquely determined. A common modelling approach is therefore to work under the objective probability measure $\mathbb{P}$ and to interpret deviations between expected future spot prices and observed forward prices as explicit risk premia. Under this interpretation, the seasonal component $s(t)$ represents the shape of expected future spot prices under the objective measure,

$$s(t) \approx \mathbb{E}_{\mathbb{P}}[S(t)],$$

while the correction term $\varepsilon(t)$ accounts for the overall price level as well as any risk premium embedded in forward prices. Because the level constant a_0 is absorbed into the spline rather than $s(t)$, the seasonal component captures only the shape of within-year variation, and $\varepsilon(t)$ must supply the price level in addition to the risk-premium adjustment.

It is natural to assume that the risk-premium component of ε varies with time to delivery (Benth et al., 2007). The reasoning behind this is that at short maturities, market participants possess substantial information about future market conditions, for example through weather forecasts and reservoir levels, whereas at longer maturities risk perceptions tend to stabilise. This motivates a correction whose rate of change flattens at long maturities, which we enforce by imposing the boundary condition

$$\varepsilon'(t_n) = 0. \tag{5.7}$$

We employ a parametric form of the seasonal component $s(t)$ specified a priori, while the correction term $\varepsilon(t)$ is constructed based on a maximum smoothness criterion such that the resulting forward curve is as smooth as possible while still being consistent with observed market prices. Denote by $C_0^2([t_0, t_n])$ the set of twice continuously differentiable functions on $[t_0, t_n]$ with zero derivative at t_n.

If we characterise the "smoothness" of a function $\varepsilon \in C_0^2([t_0, t_n])$ by its mean square second derivative, then we define the smoothest possible forward curve on $[t_0, t_n]$ as the solution to the optimisation problem

$$\min_{\varepsilon \in C_0^2([t_0, t_n])} \int_{t_0}^{t_n} [\varepsilon''(t)]^2 \, \mathrm{d}t. \tag{5.8}$$

Consequently, in addition to the smoothness criterion, the correction term needs to be twice continuously differentiable with a zero derivative at the right boundary according to (5.7). As we show in the theorem below, the class $C \subseteq C_0^2([t_0, t_n])$ of functions over which we seek a solution to (5.8) is the class of quartic polynomial splines.

Theorem 5.1 (Maximum-smoothness correction for average-based forwards) Let $t_0 < t_1 < \cdots < t_n$ be the common settlement grid induced by m observed average-based forward contracts with delivery periods $[T_i^s, T_i^e] \subseteq [t_0, t_n]$. Assume $\tau = 0$ and write the forward curve decomposition

$$f(t) = s(t) + \varepsilon(t), \qquad t \in [t_0, t_n],$$

where $s(\cdot)$ is specified a priori.

Consider the maximum-smoothness problem

$$\min_{\varepsilon \in C^2([t_0,t_n])} \int_{t_0}^{t_n} \left(\varepsilon''(t)\right)^2 \,\mathrm{d}t \tag{5.9}$$

subject to the constraint $\varepsilon'(t_n) = 0$ and for each contract i, the forward curve must be consistent with the observed contract price:

$$F_i^C = \int_{t_0}^{t_n} w_i(t)\, f(t)\,\mathrm{d}t = \int_{t_0}^{t_n} w_i \left(s(t) + \varepsilon(t)\right) \mathrm{d}t, \tag{5.10}$$

for delivery weights $w_i = 1/(T_i^e - T_i^s)$.

Then any minimiser ε^* of (5.9) is a piecewise quartic polynomial spline on the grid. Thus, there exist coefficients $(a_j, b_j, c_j, d_j, e_j)$, $j = 1, \ldots, n$ such that

$$\varepsilon^*(t) = a_j t^4 + b_j t^3 + c_j t^2 + d_j t + e_j, \quad t \in [t_{j-1}, t_j], j = 1, \ldots, n.$$

Remark: For ease of computation in the proof, the above theorem assumes end-of-period settlement, corresponding to the uniform weighting $w = 1/(T^e - T^s)$. The result extends naturally to marked-to-market contracts with the general weighting function $w(r; t)$ defined in (5.4).

We will need the following lemma in proving Theorem 5.1.

Lemma 5.2
If f and g are continuous functions, and h is integrable, then for all $a \le t \le b$,

$$\int_a^b f(t)h(t)\,\mathrm{d}t + \int_a^b g(t) \int_t^b h(v)\,\mathrm{d}v\,\mathrm{d}t = 0, \tag{5.11}$$

if and only if

$$f(t) = -\int_a^t g(v)\,\mathrm{d}v. \tag{5.12}$$

Proof: Applying Fubini's theorem to the second term in (5.11), where the region of integration is $\{(t, v) : a \le t \le v \le b\}$, gives

$$\begin{aligned}\int_a^b f(t)h(t)\,\mathrm{d}t + \int_a^b g(t) \int_t^b h(v)\,\mathrm{d}v\,\mathrm{d}t &= \int_a^b f(v)h(v)\,\mathrm{d}v + \int_a^b h(v) \int_a^v g(u)\,\mathrm{d}u\,\mathrm{d}v \\ &= \int_a^b h(v) \left[f(v) + \int_a^v g(u)\,\mathrm{d}u\right] \mathrm{d}v = 0,\end{aligned}$$

for any continuous function h. This holds if and only if (5.12) is satisfied. ■

Proof: (Proof of Theorem 5.1.) Fix the settlement grid $t_0 < t_1 < \cdots < t_n$ obtained by collecting all unique delivery dates $\{T_i^s, T_i^e\}$, $i = 1, \ldots, m$. We seek the "smoothest" correction function ε on $[t_0, t_n]$ that solves

$$\min_{\varepsilon \in C^2([t_0,t_n])} \int_{t_0}^{t_n} \left(\varepsilon''(t)\right)^2 \mathrm{d}t, \tag{5.13}$$

subject to the pricing constraints (5.10) imposed by the m observed average-based forward contract prices, together with the boundary condition

$$\varepsilon'(t_n) = 0. \tag{5.14}$$

To shorten notation, for a fixed contract i, we write

$$\alpha := T_i^s, \qquad \beta := T_i^e.$$

Integrating $\int_\alpha^\beta \varepsilon(t)\,\mathrm{d}t$ by parts, it applies that

$$\int_\alpha^\beta \varepsilon(t)\,\mathrm{d}t = \Big[t\,\varepsilon(t)\Big]_\alpha^\beta - \int_\alpha^\beta t\,\varepsilon'(t)\,\mathrm{d}t = \beta\,\varepsilon(\beta) - \alpha\,\varepsilon(\alpha) - \int_\alpha^\beta t\,\varepsilon'(t)\,\mathrm{d}t. \tag{5.15}$$

Applying integration by parts again to $\int_\alpha^\beta t\,\varepsilon'(t)\,\mathrm{d}t$ yields

$$\int_\alpha^\beta t\,\varepsilon'(t)\,\mathrm{d}t = \Big[\frac{t^2}{2}\,\varepsilon'(t)\Big]_\alpha^\beta - \int_\alpha^\beta \frac{t^2}{2}\,\varepsilon''(t)\,\mathrm{d}t = \frac{1}{2}\Big(\beta^2\varepsilon'(\beta) - \alpha^2\varepsilon'(\alpha)\Big) - \frac{1}{2}\int_\alpha^\beta t^2\,\varepsilon''(t)\,\mathrm{d}t. \tag{5.16}$$

Substituting (5.16) into (5.15) gives

$$\int_\alpha^\beta \varepsilon(t)\,\mathrm{d}t = \beta\,\varepsilon(\beta) - \alpha\,\varepsilon(\alpha) + \frac{1}{2}\Big[\alpha^2\varepsilon'(\alpha) - \beta^2\varepsilon'(\beta) + \int_\alpha^\beta t^2\,\varepsilon''(t)\,\mathrm{d}t\Big]. \tag{5.17}$$

Defining $q(t) := \varepsilon''(t)$, the objective function (5.13) becomes $\int_{t_0}^{t_n} q(t)^2\,\mathrm{d}t$. We want to express ε and ε' in terms of q. First, applying the fundamental theorem of calculus to ε' and using the boundary condition (5.14) gives

$$\varepsilon'(x) = \varepsilon'(t_n) - \int_x^{t_n} q(t)\,\mathrm{d}t = -\int_x^{t_n} q(t)\,\mathrm{d}t, \tag{5.18}$$

for $x \in [t_0, t_n]$. Then, using the fundamental theorem of calculus again to ε and substituting (5.18), we obtain

$$\varepsilon(x) = \varepsilon(t_0) + \int_{t_0}^x \varepsilon'(v)\,\mathrm{d}v = \varepsilon(t_0) - \int_{t_0}^x \int_v^{t_n} q(t)\,\mathrm{d}t\,\mathrm{d}v. \tag{5.19}$$

Rewriting (5.10) using (5.17)–(5.19), applying Fubini's theorem to switch the order of integration where needed, and using $w_i = 1/(\beta - \alpha)$, the pricing constraint can be expressed as

$$\begin{aligned}(\beta - \alpha)F_i^C - \int_\alpha^\beta s(t)\,\mathrm{d}t = {} & (\alpha - \beta)\left[\int_{t_0}^\alpha \int_u^{t_n} q(v)\,\mathrm{d}v\,\mathrm{d}u - \varepsilon(t_0)\right] \\ & - \beta \int_\alpha^\beta \int_u^{t_n} q(v)\,\mathrm{d}v\,\mathrm{d}u + \frac{1}{2}(\beta^2 - \alpha^2)\int_\beta^{t_n} q(t)\,\mathrm{d}t \\ & - \frac{1}{2}\alpha^2 \int_\alpha^\beta q(t)\,\mathrm{d}t + \frac{1}{2}\int_\alpha^\beta t^2 q(t)\,\mathrm{d}t.\end{aligned} \tag{5.20}$$

Equation (5.20) expresses the pricing constraint implied by contract i as a linear functional of q. In what follows, we write $\alpha_i := T_i^s$ and $\beta_i := T_i^e$ for the delivery boundaries of contract i. The constrained optimisation problem can now be reformulated using Lagrange multipliers as

$$\min_{q,\lambda} \mathcal{L}(q,\lambda),$$

where

$$\begin{aligned}
\mathcal{L}(q,\lambda) &= \int_{t_0}^{t_n} q^2(t)\,\mathrm{d}t + \int_{t_0}^{t_n} \sum_{i=1}^{m} \lambda_i \mathbb{1}_{\{T_i^s<t<T_i^e\}} \left[\varepsilon(t) - F_i^C + s(t)\right] \mathrm{d}t \\
&= \int_{t_0}^{t_n} q^2(t)\,\mathrm{d}t + \int_{t_0}^{t_n} \sum_{i=1}^{m} \lambda_i \Big[\mathbb{1}_{\{t>T_i^e\}} \tfrac{1}{2}(\beta_i^2 - \alpha_i^2) q(t) \\
&\quad + \mathbb{1}_{\{t<T_i^s\}}(\alpha_i - \beta_i) \int_t^{t_n} q(v)\,\mathrm{d}v \\
&\quad + \mathbb{1}_{\{T_i^s<t<T_i^e\}} \left(\tfrac{1}{2} t^2 q(t) - \tfrac{1}{2}\alpha_i^2 q(t) - \beta_i \int_t^{t_n} q(v)\,\mathrm{d}v\right) - (\alpha_i - \beta_i)\varepsilon(t_0)\Big]\,\mathrm{d}t \\
&\quad + \int_{t_0}^{t_n} \sum_{i=1}^{m} \lambda_i \mathbb{1}_{\{T_i^s<t<T_i^e\}} \left[-F_i^C + s(t)\right] \mathrm{d}t.
\end{aligned}$$

Now, if q^* is an optimal solution to the optimisation problem, it satisfies the first-order Gâteaux condition:

$$\frac{\mathrm{d}}{\mathrm{d}\delta} \mathcal{L}(q^* + \delta k(t), \lambda)\Big|_{\delta=0} = 0,$$

for any continuous perturbation function k defined on the interval $[t_0, t_n]$ such that $k(t) = q(t) - q^*(t)$.

Evaluating the first-order condition, we have

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\delta} \mathcal{L}(q^* + \delta k(t), \lambda)\Big|_{\delta=0} &= \int_{t_0}^{t_n} \Bigg[2q^*(t) + \sum_{i=1}^{m} \lambda_i \Big(\mathbb{1}_{\{t>T_i^e\}} \tfrac{1}{2}(\beta_i^2 - \alpha_i^2) \\
&\quad + \mathbb{1}_{\{T_i^s<t<T_i^e\}} \tfrac{1}{2}(t^2 - \alpha_i^2)\Big)\Bigg] k(t)\,\mathrm{d}t \\
&\quad + \int_{t_0}^{t_n} \left[\sum_{i=1}^{m} \lambda_i \Big(\mathbb{1}_{\{t<T_i^s\}}(\alpha_i - \beta_i) - \mathbb{1}_{\{T_i^s<t<T_i^e\}}\beta_i\Big)\right] \left(\int_t^{t_n} k(v)\,\mathrm{d}v\right) \mathrm{d}t \\
&= 0,
\end{aligned}$$

Applying the integral identity in Lemma 5.2, we obtain

$$\begin{aligned}
2q^*(t) &+ \sum_{i=1}^{m} \lambda_i \left[\mathbb{1}_{t>T_i^e} \frac{1}{2}(\beta_i^2 - \alpha_i^2) + \mathbb{1}_{T_i^s<t<T_i^e} \frac{1}{2}(t^2 - \alpha_i^2)\right] \\
&= -\int_{t_0}^{t} \sum_{i=1}^{m} \lambda_i \left[\mathbb{1}_{t<T_i^s}(\alpha_i - \beta_i) - \mathbb{1}_{T_i^s<t<T_i^e}\beta_i\right] \mathrm{d}v
\end{aligned}$$

for all $t \in [t_0, t_n]$. Re-arranging, we get

$$q^*(t) = -\frac{1}{4} \sum_{i=1}^{m} \lambda_i\, t^2 + \frac{1}{2} \int_{t_0}^{t} \sum_{i=1}^{m} \lambda_i \beta_i \;\mathrm{d}v + \frac{1}{4} \sum_{i=1}^{m} \lambda_i \alpha_i^2,$$

for $t \in [t_0, t_n]$. This proves that $q^*(t)$ is a quadratic polynomial. Using $q(t) = \varepsilon''(t)$, the correction function can now be written as the following fourth-order polynomial spline function:

$$\varepsilon(t) = \begin{cases} a_1 t^4 + b_1 t^3 + c_1 t^2 + d_1 t + e_1, & t \in [t_0,t_1], \\ a_2 t^4 + b_2 t^3 + c_2 t^2 + d_2 t + e_2, & t \in [t_1,t_2], \\ \vdots & \\ a_n t^4 + b_n t^3 + c_n t^2 + d_n t + e_n, & t \in [t_{n-1},t_n]. \end{cases} \tag{5.21}$$

Thus, we have derived the correction function ε. ■

Defining the parameter vector

$$x = (a_1, b_1, c_1, d_1, e_1, a_2, b_2, c_2, d_2, e_2, \dots, a_n, b_n, c_n, d_n, e_n)$$

for the coefficients of the quartic spline in (5.21), the constrained convex quadratic programming problem in (5.9) can now be formulated as

$$\min_x \int_{t_0}^{t_n} \left[\varepsilon''(t;x)\right]^2 \,\mathrm{d}t = \min_x \sum_{j=1}^{n} \int_{t_{j-1}}^{t_j} (12a_j t^2 + 6b_j t + 2c_j)^2 \,\mathrm{d}t, \tag{5.22}$$

subject to the constraints

$$\begin{aligned} (a_{j+1} - a_j)t_j^4 + (b_{j+1} - b_j)t_j^3 + (c_{j+1} - c_j)t_j^2 + (d_{j+1} - d_j)t_j + (e_{j+1} - e_j) &= 0, \\ 4(a_{j+1} - a_j)t_j^3 + 3(b_{j+1} - b_j)t_j^2 + 2(c_{j+1} - c_j)t_j + (d_{j+1} - d_j) &= 0, \\ 12(a_{j+1} - a_j)t_j^2 + 6(b_{j+1} - b_j)t_j + 2(c_{j+1} - c_j) &= 0, \end{aligned} \tag{5.23}$$

for $j = 1, \dots, n-1$, and

$$\begin{aligned} \varepsilon'(t_n, x) &= 0 \\ F_i^C &= \int_{T_i^s}^{T_i^e} w(r;t)(\varepsilon(t;x) + s(t)) \,\mathrm{d}t, \quad i = 1, \dots, m, \\ e_1 &= F_1^C - s(t_0). \end{aligned}$$

The first three sets of constraints ensure that the correction function is a piecewise quartic polynomial spline with second-degree continuity at the knots. The fourth constraint enforces the boundary condition that the derivative of the correction function is zero at the right boundary. The fifth set of constraints ensures that the forward curve is consistent with observed market prices. The last condition pins the forward curve to the first observed contract price at the valuation date, i.e., $f(t_0) = F_1^C$, using the normalisation $t_0 = 0$ so that $e_1 = \varepsilon_1(t_0)$ holds exactly. Thus, the problem has $5n$ decision variables and $3n + m - 1$ constraints.

Integrating $\varepsilon''(t)^2$ over each interval $[t_{j-1}, t_j]$ for $j = 1, \dots, n$, we can express the minimisation problem in (5.22) in the standard form of a convex quadratic program:

$$\min_x x^\top H x,$$

where

$$H = \begin{pmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & h_n \end{pmatrix}, \quad h_j = \begin{pmatrix} \frac{144}{5}\Delta_j^{(5)} & 18\Delta_j^{(4)} & 8\Delta_j^{(3)} & 0 & 0 \\ 18\Delta_j^{(4)} & 12\Delta_j^{(3)} & 6\Delta_j^{(2)} & 0 & 0 \\ 8\Delta_j^{(3)} & 6\Delta_j^{(2)} & 4\Delta_j^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\Delta_j^{(k)} = t_j^k - t_{j-1}^k.$$

Furthermore, since the constraints are linear in x, they can be expressed in the form $Ax = b$. As an example, the first three sets of constraints in (5.23) can be expressed by constructing the following blocks of the matrix A: Denoting the block index set for segment j as

$$\begin{aligned} I_j &= \{5(j-1)+1, 5(j-1)+2, 5(j-1)+3, 5(j-1)+4, 5(j-1)+5\} \\ &= \{5(j-1)+1, \dots, 5j\}, \end{aligned}$$

for $j = 1, \dots, n-1$, such that

$$x_{I_j} = x_j = (a_j, b_j, c_j, d_j, e_j)^\top,$$

and defining the vectors

$$\phi_0(t) = (t^4, t^3, t^2, t, 1)^\top, \quad \phi_1(t) = (4t^3, 3t^2, 2t, 1, 0)^\top, \quad \phi_2(t) = (12t^2, 6t, 2, 0, 0)^\top,$$

then

$$\varepsilon_j(t) = \phi_0(t)^\top x_{I_j}, \quad \varepsilon_j'(t) = \phi_1(t)^\top x_{I_j}, \quad \varepsilon_j''(t) = \phi_2(t)^\top x_{I_j}.$$

The first three sets of constraints in (5.23) can now be expressed as

$$\begin{aligned} &\phi_0(t_j)^\top x_{I_{j+1}} - \phi_0(t_j)^\top x_{I_j} = 0, \\ &\phi_1(t_j)^\top x_{I_{j+1}} - \phi_1(t_j)^\top x_{I_j} = 0, \\ &\phi_2(t_j)^\top x_{I_{j+1}} - \phi_2(t_j)^\top x_{I_j} = 0, \qquad j = 1, \dots, n-1, \end{aligned}$$

and the corresponding blocks of the matrix A can be constructed as

$$\begin{aligned} A_{3(j-1)+1, I_j} &= -\phi_0(t_j)^\top, \quad A_{3(j-1)+1, I_{j+1}} = \phi_0(t_j)^\top, \\ A_{3(j-1)+2, I_j} &= -\phi_1(t_j)^\top, \quad A_{3(j-1)+2, I_{j+1}} = \phi_1(t_j)^\top, \\ A_{3(j-1)+3, I_j} &= -\phi_2(t_j)^\top, \quad A_{3(j-1)+3, I_{j+1}} = \phi_2(t_j)^\top, \end{aligned}$$

for $j = 1, \dots, n-1$. The remaining constraints can be expressed similarly, and the resulting matrix A and vector b can be constructed accordingly. The matrix A has dimension $(3n + m - 1) \times 5n$, and the vector b has dimension $3n + m - 1$. Note that the matrix A is sparse, with a block structure corresponding to the spline segments and the constraints linking them.

A solution to the problem can be obtained by considering the unconstrained Lagrangian

$$\mathcal{L}(x, \lambda) = x^\top H x + \lambda^\top (Ax - b),$$

where $\lambda = (\lambda_1, \dots, \lambda_{3n+m-1})$ is the vector of Lagrange multipliers. The first-order optimality conditions can be expressed as a linear system, the solution of which determines the optimal parameters of the quartic spline as well as the corresponding Lagrange multipliers. Specifically, the first-order conditions are given by

$$\begin{pmatrix} 2H & A^\top \\ A & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix},$$

which can be done using standard numerical methods such as QR factorisation.

We apply the spline smoothing procedure to $m = 16$ forward contract closing prices from Trading Hub Europe (THE), the virtual trading point for natural gas in the German market and one of the most liquid hubs in Central Europe. The dataset comprises 12 monthly contracts for delivery in 2024 and 4 quarterly contracts for delivery in 2025, obtained from Fraunhofer ISE (2026) at market close on 29 December 2023. The unique delivery-boundary dates implied by these contracts define a settlement grid of $n = 16$ intervals with 17 knot points: monthly delivery-boundary dates from January 2024 through January 2025, and quarterly boundaries for April, July, October 2025 and January 2026. We use a continuously compounded risk-free rate of $r = 0.1$.

The seasonal component $s(t)$ in (5.6) is estimated from historical THE spot prices following Chapter 2. The log-price seasonal mean is parameterised as

$$m_{\text{sea}}(t) = a_0 + a_1 \sin\left(\frac{2\pi t}{365}\right) + a_2 \cos\left(\frac{2\pi t}{365}\right).$$

Because $m_{\text{sea}}(t)$ lives in log-price space, it is exponentiated before entering the forward-curve construction. Following Benth et al. (2007), the overall price level is left to be determined by the observed contracts, so the constant a_0 is absorbed into the spline correction and only the periodic shape $\exp\left(m_{\text{sea}}(t) - a_0\right)$ is retained.

To reflect the intraweek seasonality introduced in Subsection 5.1.1, the seasonal function distinguishes weekday and weekend levels with the same spread s:

$$s(t) = \begin{cases} \exp\left(m_{\text{sea}}(t) - a_0\right) + d_{\text{wd}}, & t \text{ a weekday,} \\ \exp\left(m_{\text{sea}}(t) - a_0\right) + d_{\text{we}}, & t \text{ a weekend,} \end{cases}$$

with $d_{\text{wd}} - d_{\text{we}} = s = 0.35\,\text{EUR/MWh}$, a spread consistent with the weekday–weekend differential observed across the contracts.

Figure 5.3 shows the resulting forward curve. The spline correction aligns the curve with the 16 contract prices while the seasonal component shapes the intra-year variation, yielding a globally smooth curve that rises into the 2024–2025 winter and falls into the following summer.

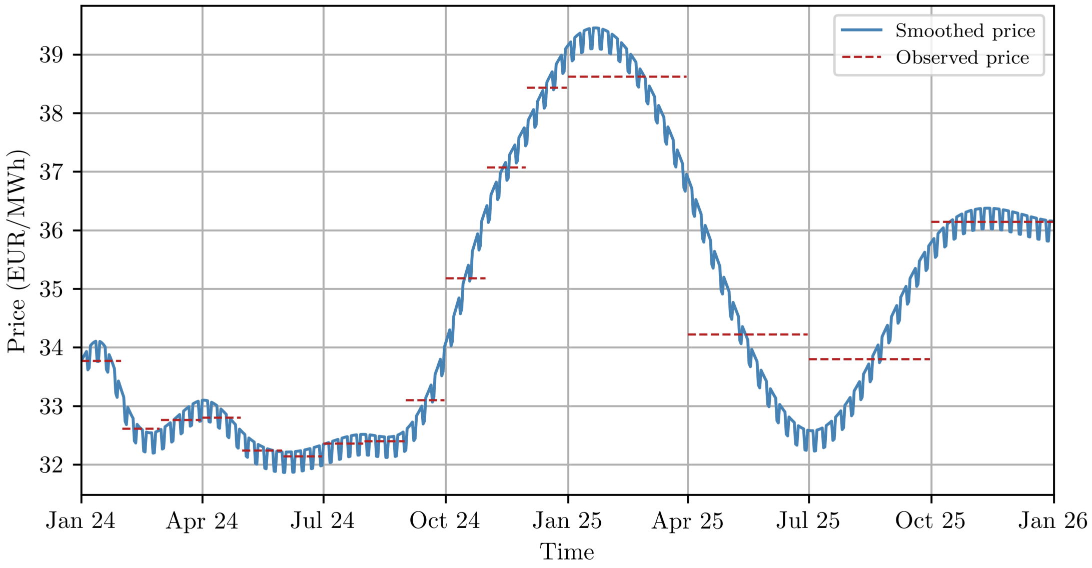


Figure 5.3: Smooth daily forward curve for THE gas (2024–2025 delivery), constructed from 12 monthly and 4 quarterly contracts using the quartic spline method of Benth et al. (2007). Separate weekday and weekend levels reflect the fixed spread $s = 0.35\,\mathrm{EUR/MWh}$.

5.2 Spot price simulation and calibration

In this section, we simulate spot price paths under the risk-neutral measure $\mathbb{Q}$ using the one-factor mean-reverting process specified in Chapter 2, and calibrate the simulation to the forward curve constructed in Section 5.1. THE day-ahead prices are obtained from Bundesnetzagentur (2026), the German Federal Network Agency responsible for regulating electricity and gas networks. The data cover daily closing prices in EUR/MWh for the period 2024–2025, matching the delivery window of the forward contracts in Section 5.1.

The mean-reversion speed κ and volatility σ are estimated under the physical measure by the AR(1) regression described in Section 2.1. The seasonal mean function $\tilde{\mu}(t)$ is parameterised using the same sinusoidal specification as in (5.6), so that a single set of seasonal parameters governs both the forward-curve construction and the physical-measure estimation. The continuous-time parameters are recovered from the OLS estimates via (2.3) and (2.4), with $\Delta t = 1$ calendar day.

The simulation is then calibrated to the forward curve following the decomposition (2.7). The deterministic mean path $m_{\mathbb{Q}}(t)$, given by (2.6), is pinned to the forward curve constructed in Section 5.1, which guarantees $\mathbb{E}_{\mathbb{Q}}[S(t)] = F(0,t)$ for all maturities t. The residual process $Y(t)$ is a zero-mean Ornstein–Uhlenbeck process driven by the estimated parameters $\hat{\kappa}$ and $\hat{\sigma}$. A total of $M = 10{,}000$ paths are generated using antithetic sampling as described in Section 2.2.

Figure 5.4 displays summary statistics of the simulated paths. The cross-sectional average closely tracks the forward curve from Figure 5.3, as expected from the calibration, while the interquartile and 5–95 % bands reflect the stochastic variability of $Y(t)$. The seasonal pattern visible in the forward curve is inherited by the simulation, and while the spread of paths widens with the horizon, it remains bounded rather than growing without limit, reflecting the mean-reverting nature of $Y(t)$.

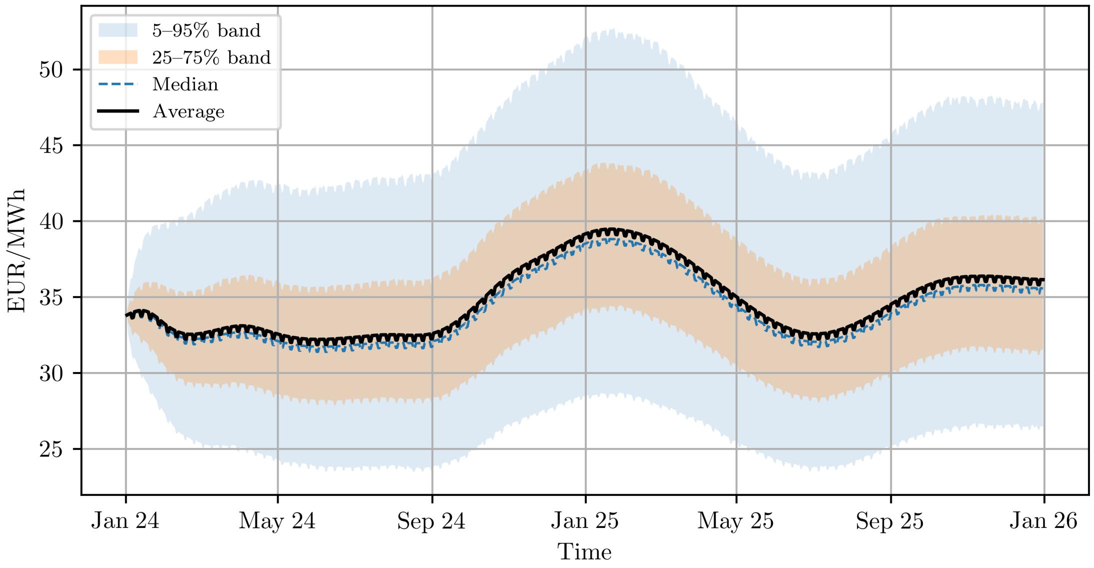


Figure 5.4: Summary statistics of $M = 10{,}000$ simulated spot price paths under the risk-neutral measure $\mathbb{Q}$. Shaded bands show the 5–95 % and interquartile ranges; the solid line is the cross-sectional average and the dashed line the median.

6 | Valuation results

This chapter presents and compares storage valuations produced by two deterministic benchmark approaches, the intrinsic and rolling intrinsic value approaches, together with six LSMC-based continuation value approximation methods, namely polynomial OLS regression, spline regression, and nodewise and joint-input variants of the MLP and KAN. The forward curve and simulated spot price paths from Chapter 5 serve as shared inputs across all methods, covering the delivery period from 2 January 2024 to 31 December 2025.

We first apply all eight methods to the storage facilities of Gas Storage Denmark (GSD). The GSD case provides a realistic setting with time- and inventory-dependent injection and withdrawal rates, a maximum working gas capacity, and binding EU regulatory filling requirements. This case study allows a direct comparison of the optimal inventory paths and estimated storage values obtained under each approach. Additionally, the specified constraints and regulatory requirements introduce non-trivial operational considerations that test the methods' ability to capture complex decision dynamics.

We then turn to a simplified storage specification with constant injection and withdrawal rates, which permits controlled convergence studies varying the number of simulated paths and the complexity of the regression basis, as well as sensitivity analyses with respect to model parameters. The main results across all models are summarised in Section 6.9.

Since the LSMC estimator derives an exercise policy from a fitted continuation value, the resulting value estimate constitutes a lower bound on the true contract value (see Proposition 3.1). A better continuation-value approximation yields a less suboptimal policy and therefore a higher estimated value, so we rank the spot-based methods by the size of their estimate throughout this chapter, with higher values indicating closer agreement with the optimal policy. We report both in-sample and out-of-sample estimates for all spot-based methods, where the out-of-sample estimate applies the policy derived from the training paths to an independent set of fresh simulation paths. A small gap between the two indicates that the continuation-value regression does not overfit the training sample.

The code is available on GitHub at `https://github.com/rsoender02/masters-thesis-gas-storage-valuation.git`.

6.1 Gas Storage Denmark's storage facilities specification

In this section, we focus on the gas storage facilities owned and operated by Gas Storage Denmark (GSD). The facilities consist of the aquifer storage facility at Stenlille in the central part of Zealand and the salt cavern facility at Lille Torup in northern Jutland (Gas Storage Denmark, 2026b). We treat these two facilities as a single storage unit with combined operational constraints and a unified inventory level, consistent with GSD's strategy of operating them in a coordinated manner.

The valuation is conducted from the perspective of the storage operator. We assume for simplicity that there are no transaction costs or bid-ask spreads, i.e., $a_1 = a_2 = b_1 = b_2 = 0$ according to (3.1), such that the unit cost of injection and unit profit of withdrawal are both equal to the spot price, i.e., $c(S(t)) = p(S(t)) = S(t)$.

As defined in Section 3.1, the maximum injection rate (IR), $i^{\max}(t, v(t))$, and withdrawal rate (WR), $w^{\max}(t, v(t))$, depend on time and the current storage fill level, as per (3.2). In our implementation, we assume a time-invariant base maximum IR of 90,720 MWh/day and a base maximum WR of 180,000 MWh/day. These are the maximum base rates specified by GSD observed on 29 December 2023, which is the last trading day before the beginning of the action period (Gas Storage Denmark, 2026a). In practice, these base rates may be subject to temporary reductions due to maintenance activities, changes in operational conditions, or changes in the technical capabilities of the storage facilities. The actual maximal base rates for injection and withdrawal during the action period from 2 January 2024 to 31 December 2025 are depicted in Figure A.4.

For example, GSD's operational data indicates that the WR was reduced by 20% for approximately two and a half months from mid January to April 2024 due to maintenance work at the Stenlille facility (Gas Storage Denmark, 2026c). To avoid look-ahead bias, we assume the base rates remain constant at their initial levels throughout the action period. Given the short duration of such reductions relative to the full two-year horizon, this is not expected to materially affect the valuation results.

The actual limits on IR and WR are determined as percentages of these base rates according to the current storage fill level, based on the specifications for SY 2024–2025 (01/04/2024 to 31/03/2025) (Gas Storage Denmark, 2026a). The corresponding stepwise availability schedules are illustrated in Figure 6.1, and we assume these schedules remain fixed throughout the action period.

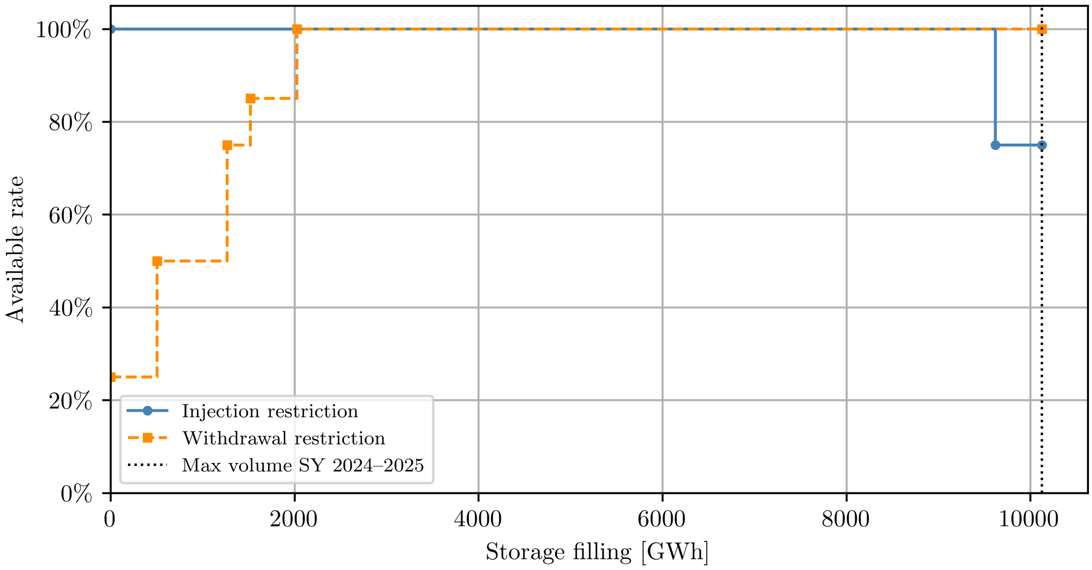


Figure 6.1: Injection and withdrawal rate limits as functions of the storage fill level specified by GSD for SY 2024–2025. The two schedules coincide over part of the fill range, while the dotted vertical line marks the maximum working gas volume for SY 2024–2025.

In addition to the constraints on injection and withdrawal rates, the storage facilities had, as of 29 December 2023, a total working gas capacity of 10,125,000 MWh. Although the actual maximum varies over the action period (see Figure A.5), we hold it fixed at this initial value.

Under the European gas storage framework, as established by the EU gas storage regulation (EU) 2022/1032, member states with underground gas storage facilities are required to ensure that storage levels reach at least 90% of total working gas capacity by 1 November each year (European Commission, 2023). This binding requirement is complemented by a set of intermediate filling targets defined along a time-dependent trajectory throughout the year, further specified in (EU) 2023/2633 and (EU) 2024/2995 (European Commission, 2023, 2024). While these intermediate targets are not legally binding in the same manner as the November obligation, they serve as benchmarks for monitoring progress and ensuring a sufficiently smooth and timely injection profile.

As such, in addition to imposing the minimum volume constraint of 90% by 1 November, we also use the intermediate targets as constraints specified in these regulations in our valuation models, ensuring that the optimal inventory paths adhere to these regulatory requirements. The intermediate targets are specified as a percentage of total working gas capacity at specific dates throughout the year, as detailed in Table 6.2.

It is worth noting that Denmark missed the 90% target in both 2024 and 2025, with storage levels reaching only around 80% and 62% of total working gas capacity, respectively, due to market conditions that did not provide sufficient incentives to fill the gas storages (Gas Storage Denmark, 2026a; Danish Energy Agency, 2025). Nevertheless, for the purposes of this valuation exercise, we assume that the storage operator fully complies with the regulatory requirements and meets the specified filling targets without deviation.

Table 6.2: Minimum storage filling requirements as a percentage of total working gas capacity. The 2024 targets follow (EU) 2023/2633, while the 2025 targets follow (EU) 2024/2995.

Anchor date	Minimum volume requirement	
	2024	2025
1 February	$\geq 45\%$	$\geq 45\%$
1 May	$\geq 40\%$	$\geq 40\%$
1 July	$\geq 60\%$	$\geq 60\%$
1 September	$\geq 80\%$	$\geq 75\%$
1 November	$\geq 90\%$	$\geq 90\%$

Rather than enforcing for example a linearly interpolated path between the regulatory anchor dates, we impose a feasibility-based lower bound. At each date, the minimum admissible inventory is defined as the smallest inventory level from which all remaining anchor requirements can still be met under the maximum feasible injection rate. This specification preserves the legally and operationally relevant obligations at the anchor dates, while avoiding the additional rigidity introduced by linear interpolation on the dates in between. In economic terms, the feasibility-based constraint allows the storage operator to hold only the inventory that is necessary to remain compliant with future requirements, instead of forcing inventory to track an arbitrary interim trajectory. The resulting minimum and maximum inventory constraints over the action period are illustrated in Figure 6.3.

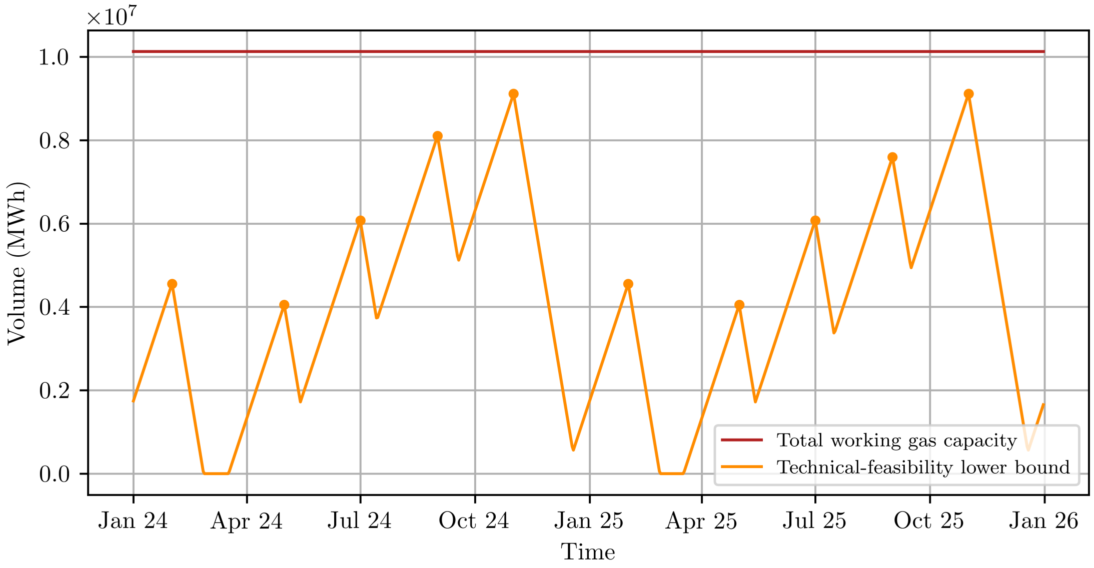


Figure 6.3: Minimum and maximum inventory constraints over time. The plot shows the minimum inventory level (in MWh) as a feasibility-based floor implied by the intermediate filling targets specified in (EU) 2023/2633 and (EU) 2024/2995, and the maximum feasible injection rate, while the maximum inventory level (in MWh) is held constant at GSD's total working gas capacity.

Finally, the initial volume level is set to 9,153,925 MWh, as this corresponds to the actual inventory level observed on 29 December 2023. For comparability, we set the terminal target volume equal to the initial volume, so that the valuation reflects trading performance over the horizon rather than gains or losses arising from a different terminal inventory level. This terminal target is enforced by imposing a large penalty on all terminal inventory levels except the target, ensuring that the optimal policy always returns to the target by the end of the action period.

6.2 Intrinsic valuation approach

In this section, we present the valuation results obtained using the intrinsic valuation approach. As explained in Subsection 1.2.1, intrinsic valuation uses the forward curve to estimate the value and assumes no random variation in the prices. The intrinsic value is calculated by discounting the expected cash flows based on the forward prices.

In our implementation, we use the forward curve constructed in Chapter 5 to determine the optimal inventory path and the corresponding cash flows. The optimal inventory path is determined by solving a backward dynamic programming problem that maximises the discounted cash flows over the action period from 2 January 2024 to 31 December 2025, subject to the storage constraints specified in Section 3.1. We choose 2 January as the starting point since this is the first trading date of the action period for which spot prices are available. The forward curve is constructed once from the closing prices observed on 29 December 2023, the last available trading date before the action period begins, and held fixed throughout.

Using the constructed forward curve, the intrinsic value of the storage facility is calculated to be approximately €78.999 million. The optimal inventory path corresponding to this intrinsic value is shown in Figure 6.4. When compared with the forward curve in Figure 5.3, the inventory path generally exhibits an inverse relationship with the quartic spline approximation of forward prices, with inventory levels increasing when forward prices are low and decreasing when forward prices are high. More specifically, the path reflects a strategy of injecting gas during the summer months of 2024, when forward prices are relatively low, followed by full withdrawal in January 2025 when forward prices are at their peak. This pattern repeats in the second year of the action period, with a second injection phase in the summer of 2025 followed by withdrawal towards the end of 2025, before injecting to reach the target level. The discontinuous, bang-bang nature of the path is characteristic of intrinsic valuation. Additionally, due to the weekday-weekend spread in the forward curve, the inventory path exhibits a sawtooth pattern, with injections and withdrawals concentrated on weekdays when prices are more favourable, and inactivity on weekends when prices are less favourable.

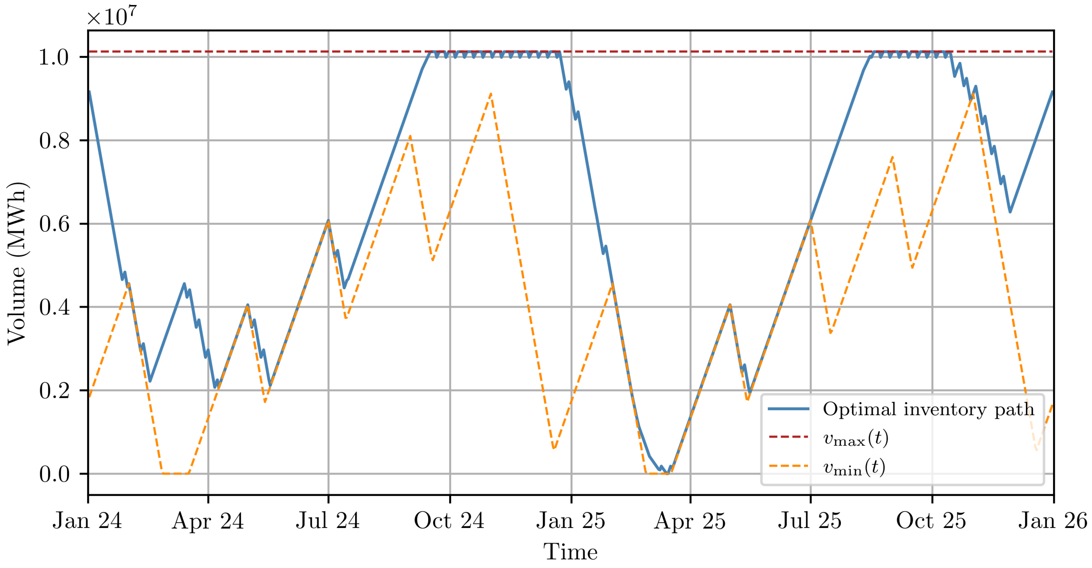


Figure 6.4: Optimal inventory path obtained from the intrinsic valuation approach of trading on the forward curve. The plot shows the volume level $v(t)$ (in MWh) over the action period from January 2024 to December 2025.

The daily inventory changes are displayed in Figure 6.5, showing that most decisions lie at the extremes of maximum feasible injection, maximum feasible withdrawal, or inactivity. Since the forward curve is fixed and deterministic, the dynamic programme generally prescribes these corner solutions whenever a spread is profitable, with little incentive to smooth transitions between states. The main deviations from this bang-bang pattern arise from the inventory-dependent injection and withdrawal rate constraints described in Section 6.1, which reduce the feasible IR and WR at certain fill levels and therefore force some intermediate daily inventory changes.

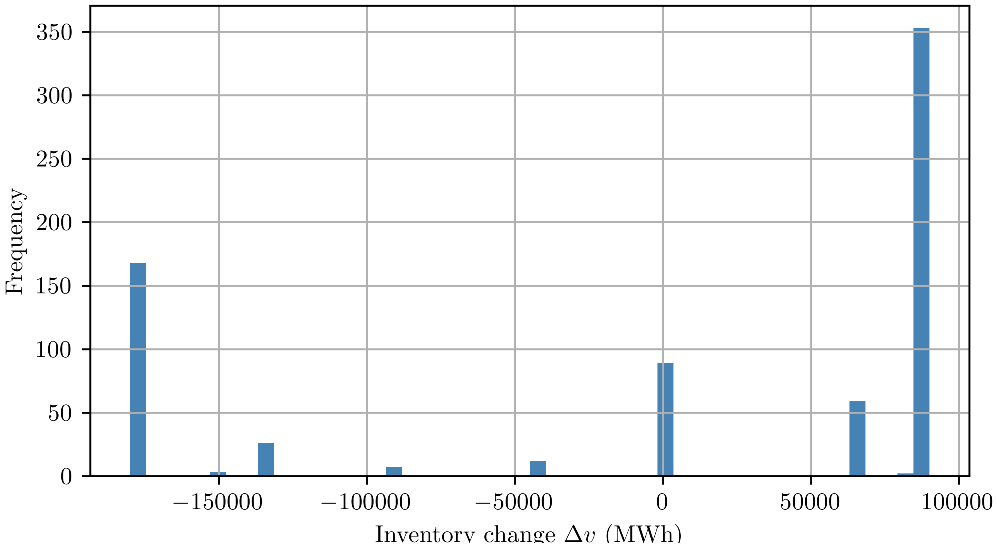


Figure 6.5: Distribution of inventory changes across simulations using the intrinsic valuation method.

The intrinsic strategy is static, since it is fixed by the forward curve observed on 29 December 2023 and cannot adjust to later price movements. The next section therefore considers the rolling intrinsic approach in Section 6.3, which re-optimises the strategy as the forward curve is updated.

6.3 Rolling intrinsic valuation approach

The static intrinsic and LSMC methods both condition on the information available at the single valuation date $t = 0$. The static intrinsic evaluates the contract against the forward curve $F(0, \cdot)$ observed on 29 December 2023, and LSMC draws simulated spot price paths from the distribution calibrated at that same origin. The rolling intrinsic instead formulates a sequential decision problem in which the information set expands at each trading date τ, and the operator conditions each re-optimisation on the forward curve $F(\tau, \cdot)$ then available.

The improvement over the static intrinsic is purely informational. The operator can act on spreads that were not present in the forward curve at $t = 0$, and can revise the schedule when previously identified spreads narrow or close. The strategy remains entirely forward-based and carries no exposure to realised spot prices, but it captures the additional value that arises from observing a sequence of forward curves rather than a single one.

At each trading date τ, a new forward curve is constructed from the contracts available on that date, and the dynamic programme is re-solved over the remaining sub-period $[\tau, T]$. Only the first decision of each re-optimisation is executed, and the programme is then discarded and re-run the following trading day. As discussed in Subsection 1.2.2, this procedure captures additional value beyond the static intrinsic strategy, since the operator can respond to changes in the forward curve rather than remaining locked into the original schedule.

Forward curves are constructed daily using the quartic spline smoothing procedure described in Chapter 5. On each trading date τ, the available set of monthly and quarterly contracts is used to fit the curve, with τ serving as the time origin ($\tau = 0$). The resulting curve is then evaluated on a daily grid from τ to the end of the last contract's delivery period.

In the forward dataset, the current month's contract remains listed through the ninth calendar day of the delivery month. From the tenth onward it rolls off and the earliest available monthly contract covers the following month. The contract set on any trading date is assembled by taking all available monthly contracts and all quarterly contracts whose delivery period is not already spanned by a monthly, sorted chronologically, with deduplication ensuring no overlapping delivery periods in the constraint matrix A.

This same roll-off also determines when forward curves are frozen. On non-trading days (weekends, public holidays, and dates after 9 December 2025) the curve is held at the most recent available value, and an analogous freeze applies from the tenth of each month through month-end. Since the earliest contract then starts on the first of the following month, the spline would otherwise extrapolate over the current month's remaining days. To avoid using these extrapolated prices, the curve is instead frozen at its value from the ninth, the last date on which the current month's contract was still listed, ensuring all decisions are based on observed market data.

Based on the constructed forward curves, the rolling intrinsic value of the storage facility is €107.245 million, which is approximately 36% higher than the static intrinsic value, consistent with the discussion in Subsection 1.2.2. However, as noted in Subsection 1.2.2, the rolling intrinsic value still constitutes a lower bound on the full value of the storage contract, since it does not account for the additional optionality arising from spot price volatility. The difference between the rolling intrinsic value and the spot-based LSMC valuations reported in Sections 6.4, 6.6 and 6.7 therefore provides an estimate of the extrinsic value.

Figure 6.6 shows the optimal inventory path obtained under the rolling intrinsic valuation. Relative to the static intrinsic path in Figure 6.4, both the seasonal injection-withdrawal pattern and the sawtooth variation are less pronounced. This reflects the daily re-optimisation of the strategy under updated forward curves, which can alter the attractiveness of previously identified trades and create new opportunities. In addition, the bang-bang nature of the path is even more pronounced than under the static approach, as seen in Figure A.6.

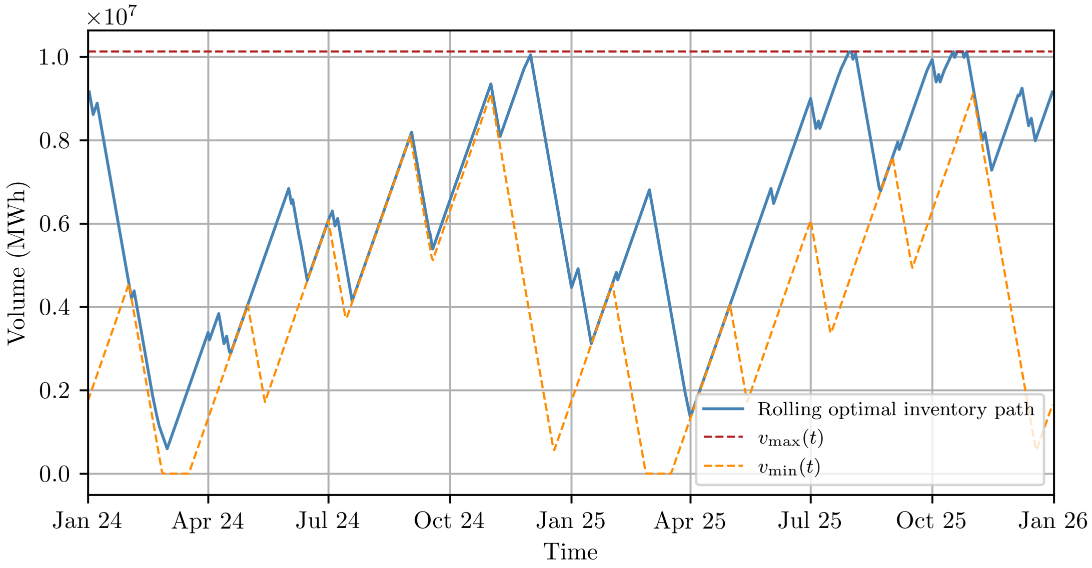


Figure 6.6: Optimal inventory path obtained from the rolling intrinsic valuation approach. The plot shows the volume level $v(t)$ (in MWh) over the action period from January 2024 to December 2025. On each day, the strategy is re-optimised using the forward curve constructed from the most recent available trading date.

6.4 OLS-based LSMC valuation

We now apply the classical LSMC approach with polynomial OLS regression to the GSD storage problem. This provides the baseline spot-based valuation against which the later MLP- and KAN-based implementations are compared. The section first describes the numerical implementation choices that make the method computationally feasible at the present scale, then examines the realised inventory paths and estimated value distribution implied by the OLS policy, and finally analyses the sensitivity of the results to the inventory-grid resolution.

6.4.1 Numerical implementation

While the mathematical LSMC algorithm is stated in Algorithm 1, the practical implementation requires several numerical choices to make the method computationally tractable at the scale considered here. The purpose of these choices is not to alter the algorithmic logic, but to reduce runtime while preserving the backward-recursive structure of the valuation. Because the backward recursion is sequential in time, it cannot be parallelised across dates. The computational gains therefore come from accelerating the operations performed within each time step. In particular, the implementation combines parallelisation and vectorised array operations in the computationally intensive steps. The vectorisation is mainly applied across the simulated price paths, for example when constructing the regression basis, evaluating discounted continuation values, and forming pathwise cash-flow arrays, while the parallelisation is used primarily over the simulated paths in the Bellman update, interpolation, and forward-rollout steps.

A second important choice is to precompute the feasible action sets for each time and inventory level before the backward pass begins. This avoids repeatedly reconstructing the admissible injection and withdrawal decisions inside the dynamic programme. Similarly, continuation values are restricted to states that remain feasible with respect to the terminal target. More precisely, at each time t, the continuation function is only estimated over inventory levels $v(t)$ for which there exists at least one admissible sequence of future injection and withdrawal decisions satisfying the operational constraints and leading to the prescribed terminal inventory $v(T)$. States from which the terminal target can no longer be reached are excluded from the approximation domain and assigned a continuation value of $-\infty$. This feasibility pruning both reduces unnecessary computation and prevents the smooth regression step from assigning finite continuation values to infeasible states.

The implementation also uses a two-level inventory grid, following the interpolation idea introduced earlier in Subsection 3.3.2. The OLS regressions are estimated on a coarse grid, after which the continuation values are interpolated to a finer grid, on which the Bellman comparison is carried out. This preserves a relatively fine decision resolution without requiring a separate regression at every fine-grid node. After the backward pass, the stored regression coefficients are used in a separate forward simulation step to generate realised inventory paths and pathwise values. This is useful both for interpreting the resulting strategy and for evaluating the realised distribution of $\hat{V}_0$. Finally, the simulation itself uses antithetic price paths and JIT-compiled functions with statically allocated working arrays, avoiding repeated memory allocation within the backward loop, further reducing the computational cost of the implementation. The specific numerical settings used in the GSD application are summarised in Table 6.7.

Table 6.7: Numerical settings for the OLS-based LSMC implementation for the GSD case.

Parameter	Value
Number of simulated paths, M	10,000
Horizon length, T	729 daily steps
Coarse-grid spacing, α_c	10,000 MWh
Refinement factor, r	4
Fine-grid spacing, α_f	2,500 MWh
Polynomial basis degree	3
Discount rate	10% p.a.
Infeasible continuation value	$C(t, v, S) = -\infty$

With these implementation choices in place, we now turn to the valuation results for the GSD case.

The OLS-based LSMC strategy preserves the same broad seasonal structure as the intrinsic and rolling intrinsic benchmarks, with inventory being reduced during high-price winter periods and rebuilt during lower-price summer periods, as shown in Figure 6.8. Relative to the static intrinsic path in Figure 6.4, the average LSMC path is less tightly pinned to the inventory bounds and therefore appears less purely bang-bang. This should, however, not be interpreted as implying smooth pathwise controls. Rather, the smoother appearance reflects averaging across many simulated paths with different realised price trajectories, while the underlying inventory decisions remain concentrated near the feasible extremes, as shown by Figure A.7.

Compared with the rolling intrinsic path in Figure 6.6, the average LSMC path follows a similar overall timing of injections and withdrawals, but with substantially greater cross-sectional dispersion, as reflected in the wide 10–90% band. This dispersion is economically natural, since the spot-based LSMC policy adapts to stochastic price realisations rather than to a single deterministic or forward-updated curve. At the same time, the visible sawtooth variation remains present in the average path, showing that the method still captures the weekday–weekend spread embedded in the forward curve and adopted in the price simulations while adding the flexibility associated with spot-price uncertainty.

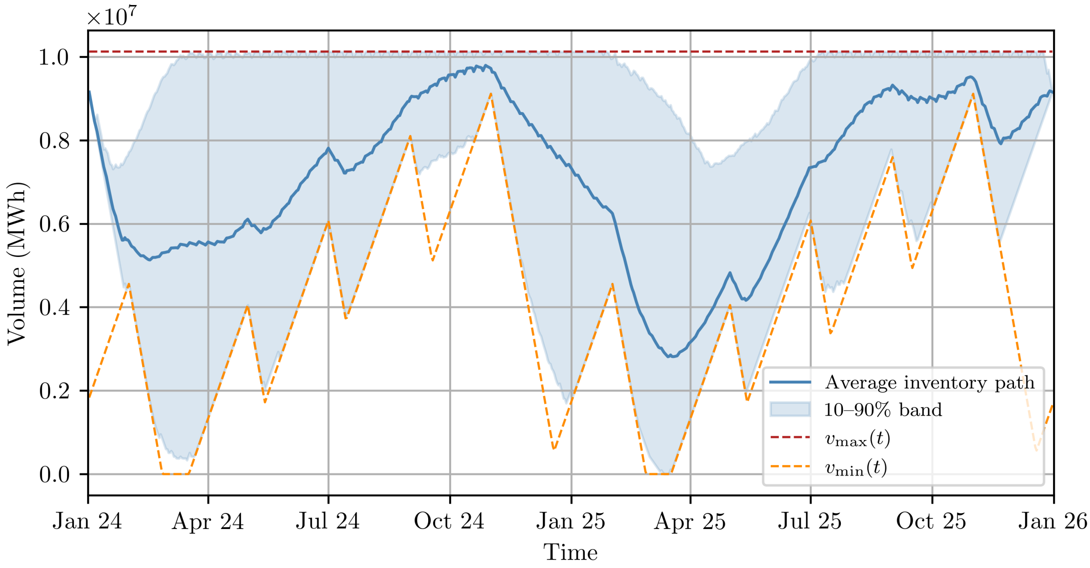


Figure 6.8: Realised inventory paths across paths using the OLS-based LSMC method. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the minimum and maximum inventory constraints.

Figure 6.9 shows that the pathwise estimates of $\hat{V}_0$ are concentrated around their mean and exhibit an approximately bell-shaped distribution. This visual impression suggests that the distribution is close to Gaussian. However, the normality tests give mixed evidence since the Shapiro–Wilk test does not reject normality at the 5% level, whereas the Jarque–Bera test rejects exact normality. The distribution of $\hat{V}_0$ is therefore best interpreted as broadly Gaussian in shape, while still exhibiting statistically detectable departures from exact normality.

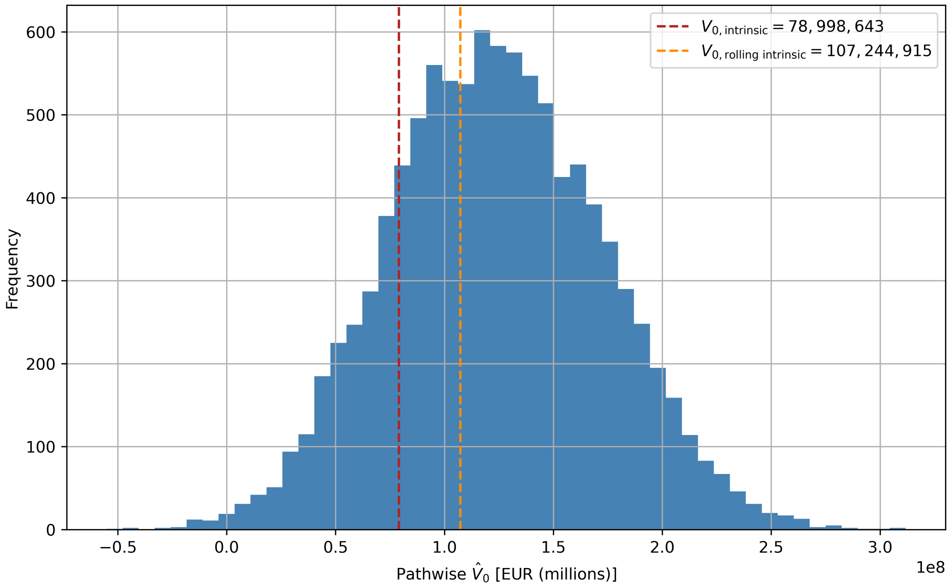


Figure 6.9: Distribution of estimated storage value $\hat{V}_0$ across simulations using the OLS-based LSMC method.

To assess how the regulatory constraints specified in Table 6.2 affect the estimated storage value, we re-run the OLS-based LSMC valuation under four alternative constraint configurations, with results reported in Table 6.10. The 2025 regulation eased the 1 September requirement from 80% to 75% relative to 2024, and reverting this to the stricter 2024 level has virtually no effect (−0.09%), confirming that the relaxation carried little economic significance. The February winter-floor constraint is more binding, as relaxing it from 45% to 20% raises the estimated value by 3.06% given that the operator is otherwise forced to forgo profitable withdrawals during the high-price winter period. Relaxing the November target from 90% to 80% yields a modest 1.55% gain, while removing all intermediate requirements entirely raises the value by 32.73%, illustrating the material economic cost of the EU regulatory framework as a whole.

Table 6.10: Sensitivity of the OLS-based storage value estimate to alternative intermediate filling requirements. Each scenario modifies one anchor-date target (or all) relative to the baseline specification in Table 6.2. Values are in millions of euros (€M).

Scenario	$\hat{V}_0$ (€M)	Relative to baseline
Baseline	123.481	—
1 September: 75% → 80%	123.365	−0.09%
1 February: 45% → 20%	127.264	+3.06%
1 November: 90% → 80%	125.396	+1.55%
No intermediate requirements	163.890	+32.73%

6.4.2 Inventory grid convergence

As discussed in Subsection 3.3.1 and Subsection 6.4.1, the inventory state space is discretised into a uniform grid of equal-width intervals, with separate OLS regressions carried out at each grid node. In our implementation, we employ a two-level grid structure. The OLS regressions are estimated at N_c coarse-grid nodes with spacing α_c (in MWh), covering the full feasible inventory range $[0, V_{\max}]$. The resulting continuation values are then linearly interpolated to a fine grid with spacing $\alpha_f = \alpha_c/r$, where $r = 4$ is a fixed refinement factor, yielding $N_f = (N_c - 1) \cdot r + 1$ fine-grid nodes. The Bellman update and all subsequent decision evaluations are carried out on the fine grid, so that the effective resolution of the action space is α_f.

The remaining numerical choice is the coarse-grid resolution N_c and the refinement factor r. The motivation is to choose N_c large enough that interpolation error is negligible, but still small enough that the MLP- and KAN-based methods remain computationally feasible. To assess this trade-off, we carry out an OLS grid convergence study over increasingly coarse grids while holding the remaining settings fixed. The grid configurations examined are listed in Table 6.11; the refinement factor is held fixed at $r = 4$ throughout, and the finest grid ($\alpha_c = 10{,}000$ MWh) serves as the benchmark, corresponding to a fine-grid configuration with $N_c = 1{,}013$. The results are shown in Figure 6.12.

Table 6.11: Coarse-grid configurations used in the OLS grid convergence study for the GSD case. The refinement factor is fixed at $r = 4$.

α_c (MWh)	N_c
10,000	1,013
15,000	676
20,000	507
30,000	339
40,000	254
60,000	170
80,000	128

The estimated value $\hat{V}_0$ shows a noticeable but non-monotone dependence on N_c. Relative to the finest-grid benchmark, the coarsest grids understate the value materially, while the finer grids lie closer to the benchmark but still fluctuate around it. This pattern suggests that both grid resolution and Monte Carlo variation matter. The coarsest grids are too sparse to recover the value accurately, whereas among the finer grids the remaining differences are no longer monotone. One plausible reason is that the time-varying lower feasibility boundary in the storage problem has a stepwise profile, so small changes in the coarse-grid spacing can alter how the coarse nodes are positioned relative to the feasible region. This affects which inventory levels are represented explicitly in the regression and may therefore contribute to the residual variation, alongside simulation noise.

The runtime panel in Figure 6.12 clarifies why this matters more for MLP and KAN than for OLS. In the OLS implementation, the Bellman update on the fine grid remains the dominant cost, so coarsening α_c yields only modest savings. For the neural-network methods, by contrast, each coarse-grid node corresponds to a separate training problem, so reducing N_c directly reduces the computational burden.

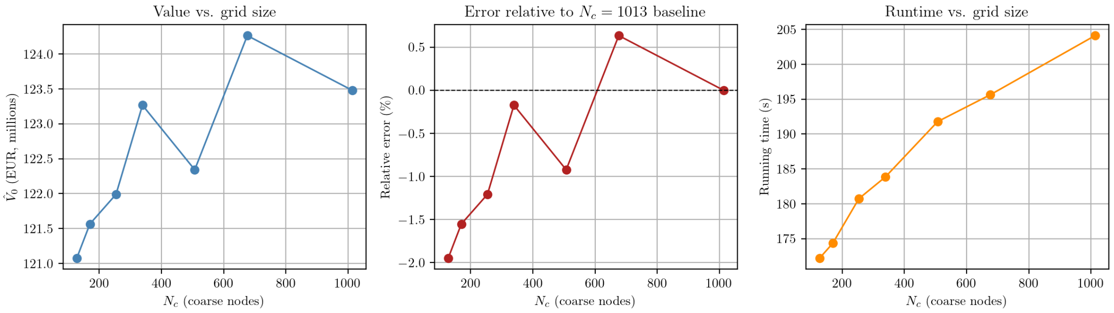


Figure 6.12: Grid convergence of the OLS-based LSMC method for the GSD case. Left: estimated storage value $\hat{V}_0$ as a function of N_c. Centre: valuation error relative to the finest-grid benchmark. Right: wall-clock runtime.

To separate interpolation error from the broader grid effect, we also examine the continuation value function $C(t, v, S)$ as a function of inventory at a fixed spot-price slice S_{50}, where S_{50} denotes the median spot price. We then measure the error induced by interpolating from progressively sparser subgrids. The interpolation errors in Figure 6.13 are negligible throughout, increasing smoothly as the subgrid becomes sparser, which is consistent with interpolation of a continuation function that is close to affine in the interior of the feasible region. The cross-sections in Figure 6.14 are visually almost indistinguishable except near the lower feasibility boundary, where the regulatory constraint introduces a sharper gradient. This indicates that the more visible variation in $\hat{V}_0$ across coarse grids in Figure 6.12 cannot be attributed mainly to interpolation alone, but rather to the effect of solving the dynamic programme on a coarser set of regression nodes.

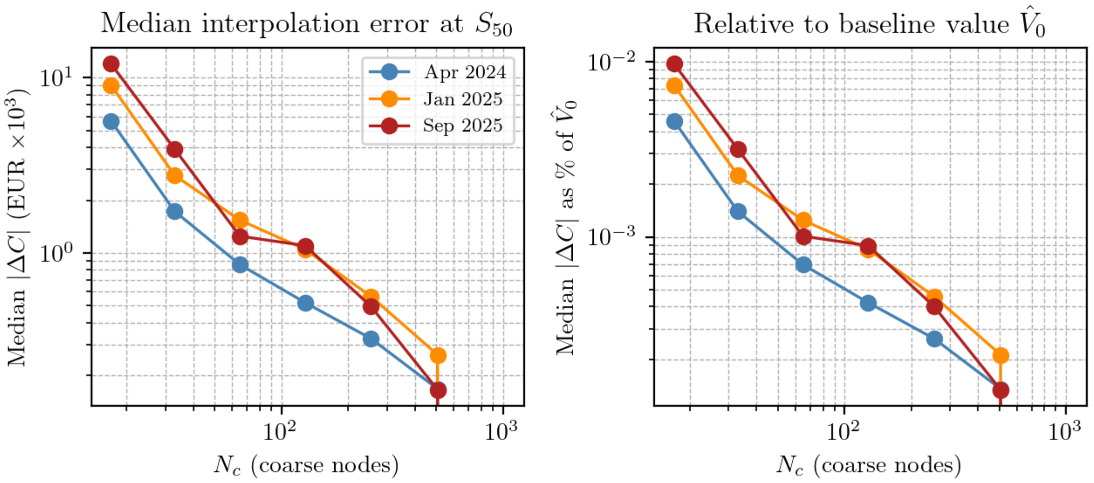


Figure 6.13: Median interpolation error in $C(t, v, S_{50})$ for the GSD case, where S_{50} denotes the median spot price. Left: absolute error in EUR. Right: error relative to $\hat{V}_0$. Each curve corresponds to a representative date.

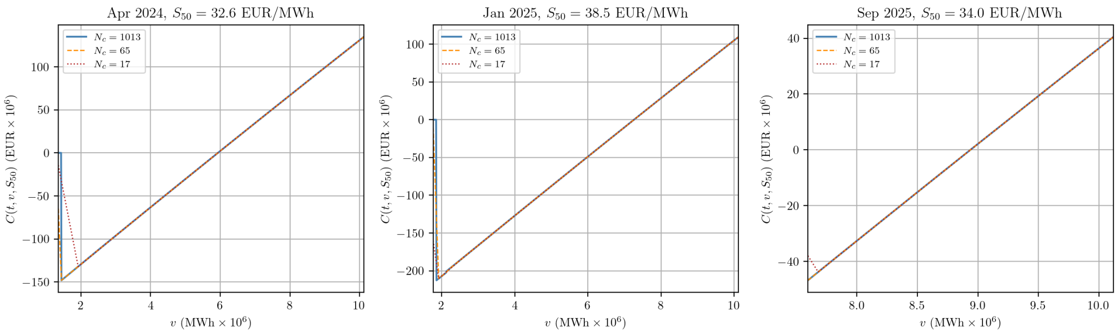


Figure 6.14: Cross-sections of $C(t, v, S_{50})$ for the GSD case at three representative dates, where S_{50} denotes the median spot price. Each panel compares the finest-grid benchmark with coarser interpolated approximations.

On the basis of these results, we retain a coarse node spacing of $\alpha_c = 40{,}000$ MWh, corresponding to $N_c = 254$ coarse nodes, for the MLP- and KAN-based implementations in the following sections. This choice is motivated by computational tractability rather than by exact grid convergence, since Figure 6.12 shows that some discretisation error remains at this resolution, but the interpolation analysis indicates that it is not driven by interpolation itself, and the reduction in the number of neural-network training calls remains important for keeping the MLP- and KAN-based valuations computationally feasible.

In addition to varying the number of coarse-grid nodes N_c, we also tried varying the refinement factor r. When testing $r \in \{2, 4, 7, 16, 32, 64\}$ across coarse-grid spacings $\alpha_c \in \{10{,}000, 2{,}500, 1{,}000\}$ MWh, the value estimates were identical. This is consistent with the evidence in Figures 6.13 and 6.14 that linear interpolation already provides an accurate approximation of the continuation value on the fine grid, since $C(t, v, S)$ is close to affine in inventory over most of the feasible region. It is also in line with the policy behaviour discussed earlier, as the distribution of inventory changes in Figure A.7 shows that pathwise actions remain concentrated near the feasible extremes. Taken together, these findings suggest that, for the GSD case, increasing the refinement factor does not unlock additional economically relevant flexibility, and that the remaining grid sensitivity is driven mainly by the number and placement of the coarse regression nodes rather than by the interpolation step.

6.5 Spline-based LSMC valuation

The spline-based LSMC valuation, using three internal knots and a power basis of $\{1, x, x^2, x^3\}$, yields an estimated storage value of €124.247 million, which is 0.62% above the OLS benchmark. We verified that with no interior knots, the value estimate is identical to the OLS benchmark. The marginal gain in using splines suggests that the continuation value does not depart substantially from a polynomial form over most of the feasible region. The realised inventory paths under the spline policy are shown in Figure A.8 and exhibit a similar structure to those produced by the OLS policy, consistent with the small difference in value estimates.

6.6 MLP-based LSMC valuation

Unless otherwise stated, the MLP-based method follows the same overall LSMC workflow and inventory-state discretisation as the OLS implementation, with the polynomial regression step replaced by a neural-network approximation. We use a coarser grid of $\alpha_c = 40{,}000$ MWh with refinement factor $r = 16$, such that the grid spacing remains the same as in the OLS benchmark, namely $\alpha_f = 2{,}500$ MWh on the fine grid.

We use a relatively small MLP specification with two hidden layers of 64 hidden units each, using ReLU activations, trained at each grid node using the L-BFGS optimiser with at most 30 quasi-Newton iterations. Both the spot-price input and the continuation-value target are standardised using z-scores before training, and the network output is rescaled back to the original units for evaluation. These choices were dictated by the computational burden of the problem. With $T = 729$ time steps and $N_c = 254$ coarse nodes, up to $729 \times 254 \approx 185{,}000$ networks must be trained during the backward pass, so larger architectures or longer training runs would make the overall valuation computationally expensive. Because L-BFGS exploits curvature information, it converges faster per iteration than first-order methods, which partly compensates for the small iteration budget. The drawback is that the network has less capacity and less opportunity to converge fully at each node, which may weaken the quality of the continuation-value approximation relative to the OLS benchmark.

Indeed, the MLP-based LSMC valuation yields an estimated storage value of €98.856 million. This remains above the intrinsic benchmark but falls below the rolling intrinsic benchmark, and is materially below the OLS-based valuation by 20%, suggesting that the continuation-value approximation is less effective in the present specification.

The realised inventory paths under the MLP policy are shown in Figure 6.15. The MLP-based policy preserves the same broad seasonal structure as the OLS policy, with inventory being accumulated in lower-price periods and reduced in higher-price periods. The average path also displays some weekday–weekend variation, but the sawtooth pattern is less pronounced than under OLS. At the same time, the wide 10–90% band indicates that the policy still responds substantially to pathwise spot-price realisations. Relative to Figure 6.8, the average path appears somewhat smoother and less tightly aligned with the short-run variation in the operational bounds, which is consistent with the lower value estimate.

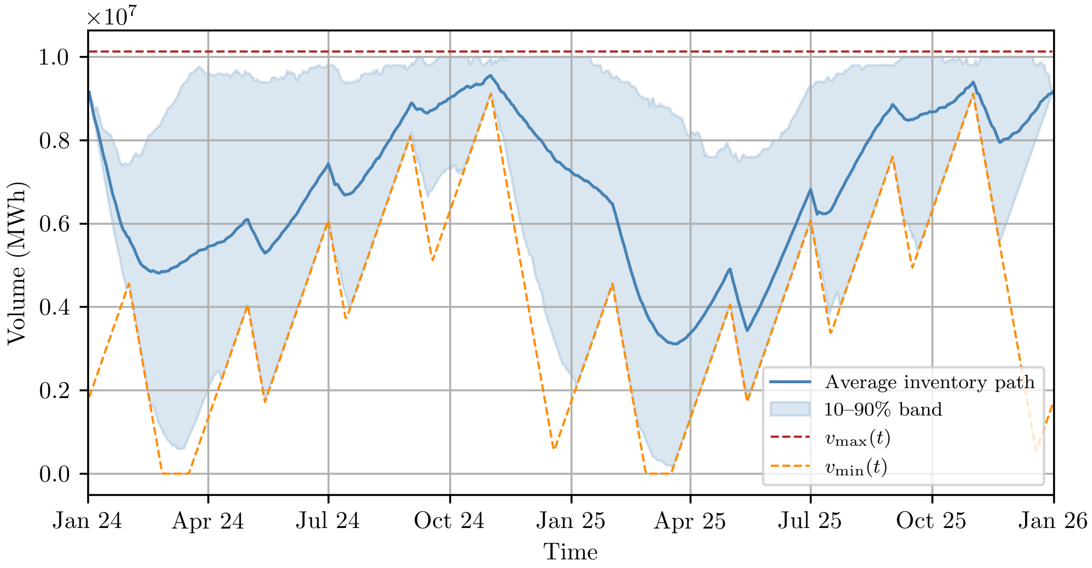


Figure 6.15: Realised inventory paths across paths using the MLP-based LSMC method. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the minimum and maximum inventory constraints.

Additionally, the average inventory path under the MLP policy remains closer to the lower feasibility boundary than under OLS. This is consistent with the nodewise training structure, which fits a separate network at each coarse-grid node and therefore cannot pool information across inventory levels. This limitation is particularly costly near the lower feasibility boundary, where the continuation value has a sharper gradient across inventory levels that a nodewise network, restricted to a univariate function of the spot price, cannot represent directly. The improvement seen under the joint-input MLP in the following section, which shares a single network across all inventory levels, supports this interpretation directly. Taken together, the lower value estimate and the path behaviour suggest that the MLP specification is not sufficiently expressive to match the performance of the OLS benchmark in the present configuration.

6.6.1 MLP with joint inputs

The evidence from Subsection 6.4.2 suggests that the continuation value is close to affine in inventory over most of the feasible region. This motivates a modified MLP specification in which inventory is treated as an explicit input to a single shared network, rather than fitting a separate network at each coarse-grid node. In the present setting, this change is attractive for two reasons. First, it allows the model to share information across inventory levels. Specifically, rather than treating each level as an independent univariate regression problem, a single joint network directly represents how the continuation value varies with both spot price and inventory. Second, if the dependence on v is indeed simple, a shared network can represent it with little additional complexity, while avoiding the instability that may arise when many small networks are trained separately.

Under this alternative specification, the MLP-based LSMC valuation increases to €107.101 million. This is 8.34% above the baseline MLP estimate and closer to the OLS benchmark, while still remaining slightly below it. The improvement is consistent with the interpretation above. When the inventory dependence is handled jointly rather than node by node, the continuation-value approximation directly represents the cross-inventory structure of C rather than inferring it through separate univariate fits at each level. At the same time, the remaining gap to OLS suggests that the shared-network formulation does not fully eliminate approximation error in the present configuration.

Figure A.9 supports this interpretation. The policy retains the same broad seasonal timing as in the other specifications, with injections concentrated in lower-price periods and withdrawals occurring when prices are higher. Relative to Figure 6.15, however, the average path spends less time near the lower feasibility boundary than the baseline MLP specification and is able to capture more of the weekday-weekend variation. The higher value estimate and the improved path behaviour therefore suggest that the shared-input MLP is better suited to the present problem than the nodewise MLP benchmark.

6.7 KAN-based LSMC valuation

Likewise, the KAN-based method retains the same LSMC workflow and state discretisation, differing from the OLS benchmark primarily in the functional form used to approximate the continuation value. Here too, the GSD problem forces a restrictive specification in order to keep the computation manageable. We again use $\alpha_c = 40{,}000$ MWh and $r = 16$. Due to the high computational cost of training a separate KAN at each coarse-grid node, we use a small KAN architecture with hidden width 4, grid size 3, meaning three spline intervals (equivalently, four knots) per learned univariate activation, spline degree $k = 3$, corresponding to cubic splines, only 5 optimisation steps, no regularisation penalty. These settings substantially limit both the expressive flexibility of the network and the extent to which it can be trained, so the resulting continuation-value approximation is expected to be noticeably weaker than in a less constrained computational setup.

The KAN-based LSMC valuation yields an estimated storage value of € 88.619 million. This is far below the OLS, MLP, and rolling intrinsic valuations, which indicates that the present KAN specification fails to recover a competitive continuation-value approximation for the GSD case.

The realised inventory paths under the KAN policy are shown in Figure 6.16. The average inventory path remains much closer to the lower feasibility boundary than in the OLS and MLP cases, suggesting that the KAN-based policy underutilises the flexibility of the storage asset. Although the broad seasonal pattern is still visible, the policy appears to react less aggressively to favourable trading opportunities and spends long periods in relatively low-inventory states. The narrower path variation compared with the OLS figure also points to a less responsive policy overall. Taken together, the low value estimate and the path behaviour suggest that this particular KAN configuration is not sufficiently expressive or well-trained to match the performance of the polynomial OLS benchmark.

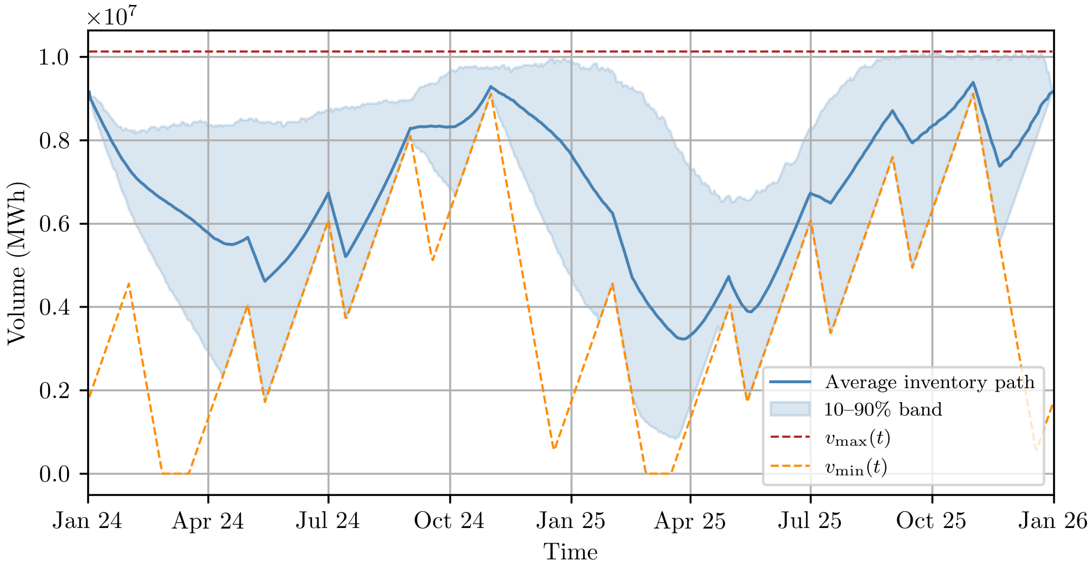


Figure 6.16: Realised inventory paths across paths using the KAN-based LSMC method. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the minimum and maximum inventory constraints.

6.7.1 KAN with joint inputs

As for the MLP case, the near-affine dependence of the continuation value on inventory motivates a KAN specification with inventory included as a joint input rather than handled through separate nodewise models. Under this alternative specification, the estimated storage value rises to €94.081 million, which is 6.1% above the baseline KAN estimate. This is a slight improvement, but the valuation remains materially below the OLS and MLP benchmarks, indicating that the shared-input formulation alleviates part of the weakness of the original KAN specification without making it competitive in the present setting. Figure A.10 is consistent with this interpretation, as the realised path appears more responsive and less persistently constrained near the lower feasibility boundary than in the baseline KAN case, though still less effective than the corresponding MLP policies. Interestingly, the path variation appears more narrow than in the baseline KAN case, which may reflect a more responsive policy, which is also seen in the more jagged average path. The average path, however, still remains close to the lower feasibility boundary for much of the action period.

6.8 Simulation study

To complement the GSD case study, we also consider a simplified Dutch salt cavern contract inspired by Boogert and De Jong (2007). The purpose of this simulation study is not to match a specific current storage product, but to work with a smaller and more stylised problem in which numerical effects can be examined more transparently. Because the storage capacity is much lower and the injection and withdrawal limits are constant, the setting is well suited for robustness checks of the OLS-based LSMC method with respect to sample size, basis complexity, and grid resolution.

6.8.1 Baseline specification and results

The key contract parameters are summarised in Table 6.17. Unless otherwise stated, the results in this section use the same spot-price model and discounting framework as in the GSD case.

Table 6.17: Parameters of the Dutch salt cavern storage contract.

Parameter	Value
Initial inventory, $v(0)$	100,000 MWh
Terminal target, $v(T)$	100,000 MWh
Working gas capacity, $V_{\max}$	250,000 MWh
Injection rate cap, $i^{\max}$	2,500 MWh/day
Withdrawal rate cap, $w^{\max}$	7,500 MWh/day
Discount rate, r	10% p.a.

6.8.2 Numerical convergence and robustness

We first examine how the OLS-based valuation responds to the number of simulated paths and to the complexity of the polynomial basis. We then compare in-sample and out-of-sample estimates to assess robustness, consider alternative polynomial families, and finally analyse the role of the inventory-grid resolution.

Figure 6.18 shows that the estimated value decreases and stabilises as the number of simulated paths increases. At low sample sizes the uncertainty bands are wide, indicating substantial Monte Carlo variation, but they narrow quickly as the sample grows. The mean estimate levels off once the number of paths reaches roughly 10,000, consistent with the choice made in the previous sections, and suggests that this order of magnitude is sufficient to obtain a reasonably stable OLS estimate in the simplified setting.

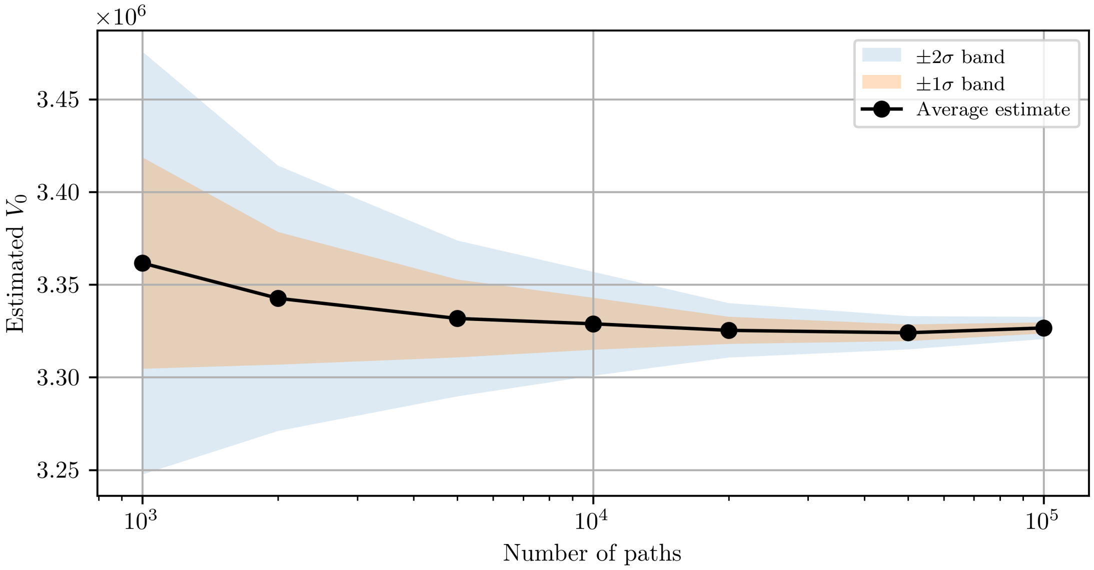


Figure 6.18: Estimated storage value $\hat{V}_0$ as a function of the number of simulated paths in the Dutch salt cavern study. The solid line shows the mean estimate, while the shaded bands indicate $\pm 1\sigma$ and $\pm 2\sigma$ intervals across 10 runs.

A similar stabilisation pattern emerges in the basis dimension, as shown in Figure 6.19. Low-degree polynomials underfit the continuation value and produce lower value estimates, while the improvements from increasing the degree become very small beyond about degree five. A moderate polynomial basis therefore appears sufficient to capture most of the relevant continuation-value variation.

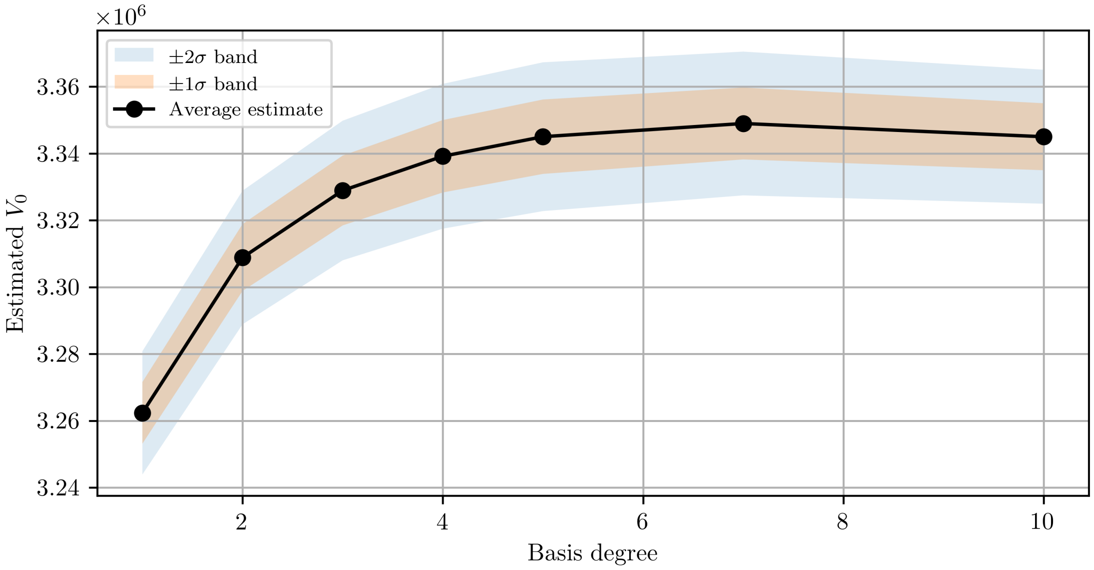


Figure 6.19: Estimated storage value $\hat{V}_0$ as a function of the polynomial basis degree in the Dutch salt cavern study. The solid line shows the mean estimate, while the shaded bands indicate $\pm 1\sigma$ and $\pm 2\sigma$ intervals across 10 runs.

Figure 6.20 complements this result by comparing in-sample and out-of-sample valuations across path counts. At low sample sizes, the in-sample estimate lies visibly above the out-of-sample estimate, indicating an upward bias from fitting and evaluating on the same paths. This gap narrows steadily as the sample size grows. Figure 6.21 reveals a similar pattern across basis degrees, where both estimates increase before flattening out. The out-of-sample estimate remains slightly below the in-sample estimate throughout, and the gap widens again at degree 10, suggesting overfitting at the highest-degree specification. Together, the two figures support the view that the OLS policy becomes more reliable once the regression sample is sufficiently large and the basis is not overly rich.

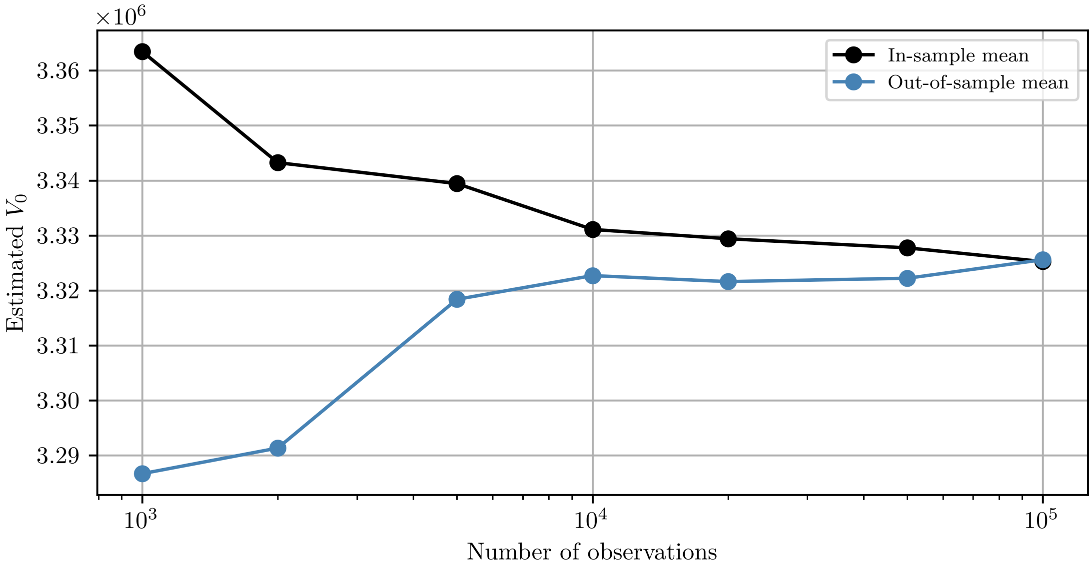


Figure 6.20: In-sample and out-of-sample OLS value estimates as a function of the number of simulated paths in the Dutch salt cavern study.

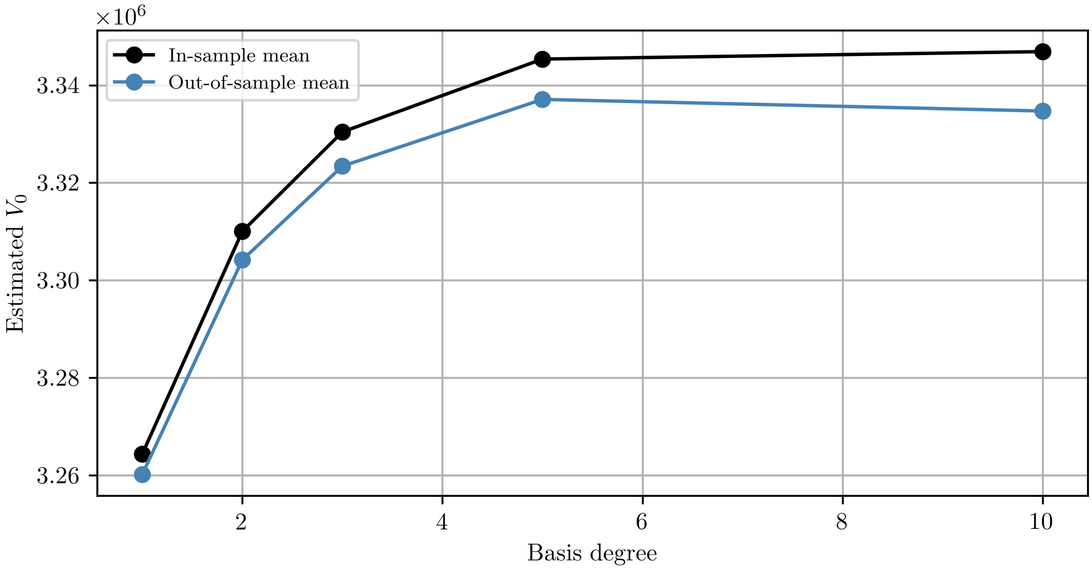


Figure 6.21: In-sample and out-of-sample OLS value estimates as a function of the polynomial basis degree in the Dutch salt cavern study.

We also analyse the sensitivity of the value estimate to the basis degree and number of knots when using the B-spline-based approach. Figure 6.22 shows that the estimated value generally increases with both degree and number of internal knots before stabilising, similar to the pattern observed in Figures 6.18 and 6.21. The effect from increasing the number of internal knots is slightly more pronounced than that from increasing the polynomial degree. The right panel shows that the difference between the in-sample and out-of-sample estimates remains negligible across most of the parameter space. Only the most flexible configurations, combining high polynomial degree with many internal knots, exhibit a visibly wider gap between the in-sample and out-of-sample estimates. As in the left panel, the gap widens marginally more when increasing the number of internal knots than when increasing the polynomial degree by the same number.

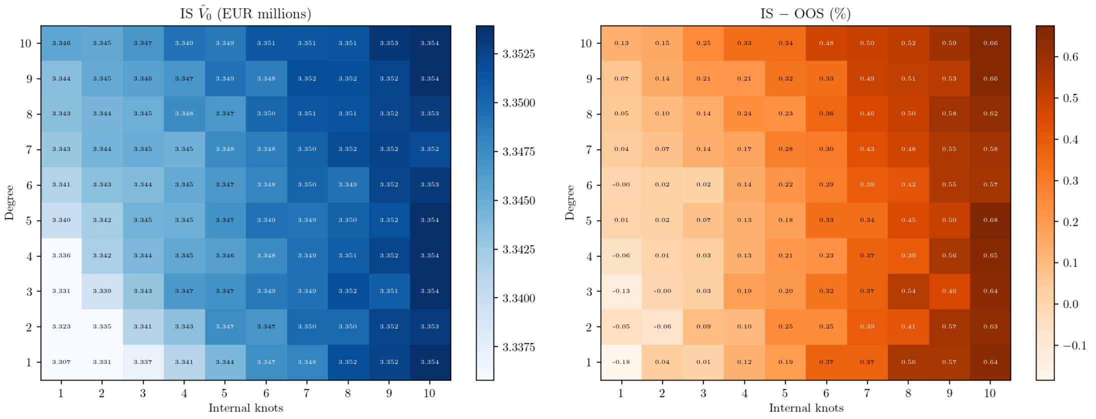


Figure 6.22: In-sample estimated value $\hat{V}_0$ (left) and in-sample minus out-of-sample discrepancy (right) across spline basis configurations using 10,000 simulated paths. Each cell corresponds to a combination of polynomial degree (vertical axis) and number of internal knots (horizontal axis). The IS−OOS discrepancy measures the degree of overfitting: positive values indicate that the in-sample value exceeds the out-of-sample value.

We now turn to the choice of polynomial family. Figure 6.23 shows that basis choice matters mainly when the approximation space is very restricted. At low degrees, the different polynomial families yield visibly different value estimates, but as the degree increases they converge towards nearly the same valuation. Taken together, Figures 6.18 to 6.21 and 6.23 indicate that the OLS method is reasonably robust in this simplified problem once the number of paths is sufficiently large and the basis is not too restrictive, consistent with the results reported by Boogert and De Jong (2007) and Longstaff and Schwartz (2001).

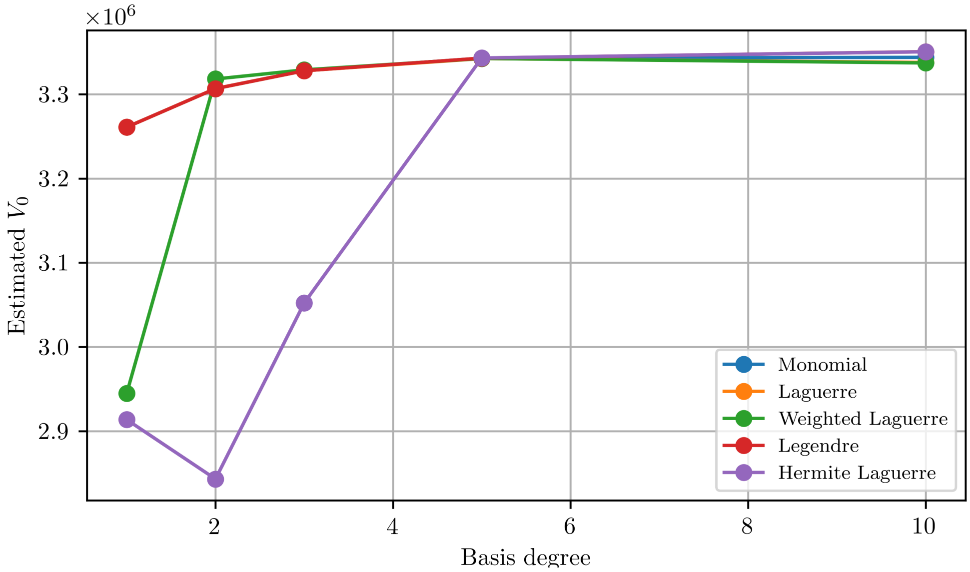


Figure 6.23: Estimated storage value $\hat{V}_0$ under alternative polynomial basis families as a function of basis degree in the Dutch salt cavern study.

Grid convergence for the OLS-based LSMC method

We apply the same convergence analysis to the Dutch salt cavern contract as done for the GSD case in Subsection 6.4.2. Since the working-gas capacity is much smaller than in the GSD case, the valuation is correspondingly more sensitive to the coarse-grid resolution. Thus, a given value of α_c represents a much larger fraction of the feasible inventory range. We therefore compare the OLS value estimate across a sequence of increasingly coarse grids, listed in Table 6.24, using $\alpha_c = 1{,}000$ MWh as the reference, corresponding to a coarse-grid configuration with $N_c = 251$. The results are shown in Figure 6.25.

Table 6.24: Coarse-grid configurations used in the OLS grid convergence study for the Dutch salt cavern contract. The refinement factor is fixed at $r = 4$.

α_c (MWh)	N_c
1,000	251
2,000	126
3,000	84
5,000	51
8,000	32
10,000	26
15,000	18
20,000	13

As in the GSD case, the value estimates are non-monotone in N_c, but the coarsest grids now produce a more pronounced downward bias. Once the grid is sufficiently refined, the estimates stabilise close to the fine-grid benchmark, though residual fluctuations remain. The runtime also increases approximately linearly with N_c as before.

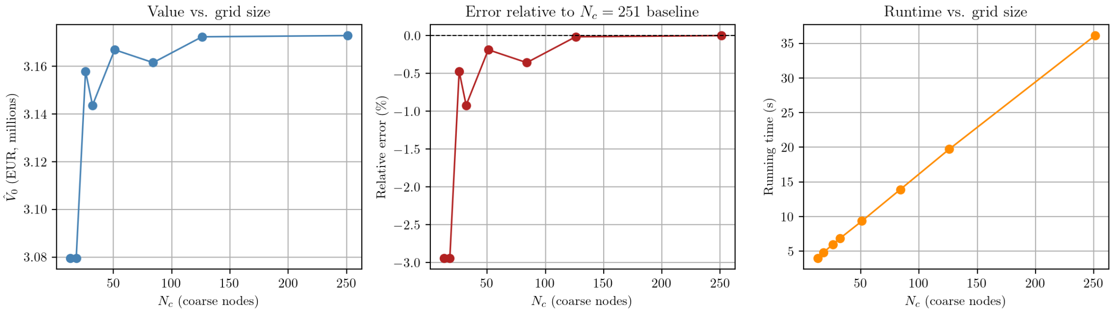


Figure 6.25: Grid convergence of the OLS-based LSMC method for the Dutch salt cavern contract. Left: estimated storage value $\hat{V}_0$ as a function of N_c. Centre: valuation error relative to the finest-grid benchmark. Right: runtime as a function of N_c.

To understand the source of the bias at coarser grids, we repeat the continuation-value shape analysis of Subsection 6.4.2. Starting from the $N_c = 251$ OLS run, we subsample the coarse grid, interpolate back to the fine grid, and measure the resulting interpolation error.

The cross-sections of $C(t, v, S_{50})$ in Figure 6.26 reveal that $C(t, v, S_{50})$ is essentially exactly linear in v across all three representative dates, and the curves for different subsampling levels are indistinguishable. Consequently, the interpolation errors in Figure 6.27 are negligible throughout and increase only marginally as the subgrid becomes sparser.

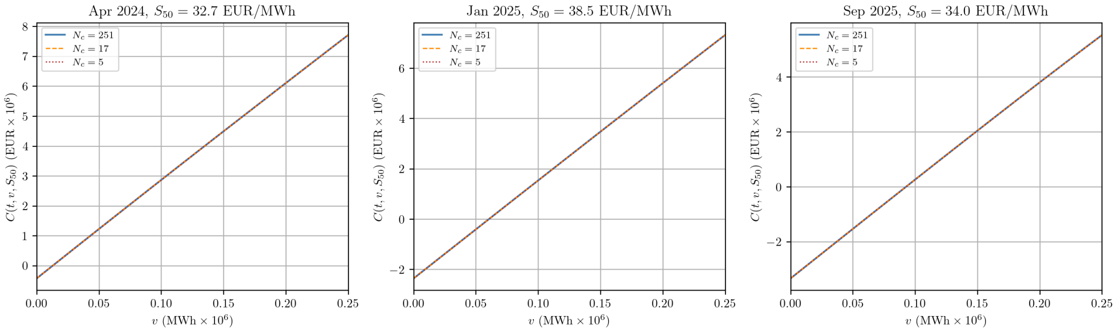


Figure 6.26: Cross-sections of $C(t, v, S_{50})$ for the Dutch salt cavern contract at three representative dates, where S_{50} denotes the median spot price. Each panel compares the finest-grid benchmark with coarser interpolated approximations.

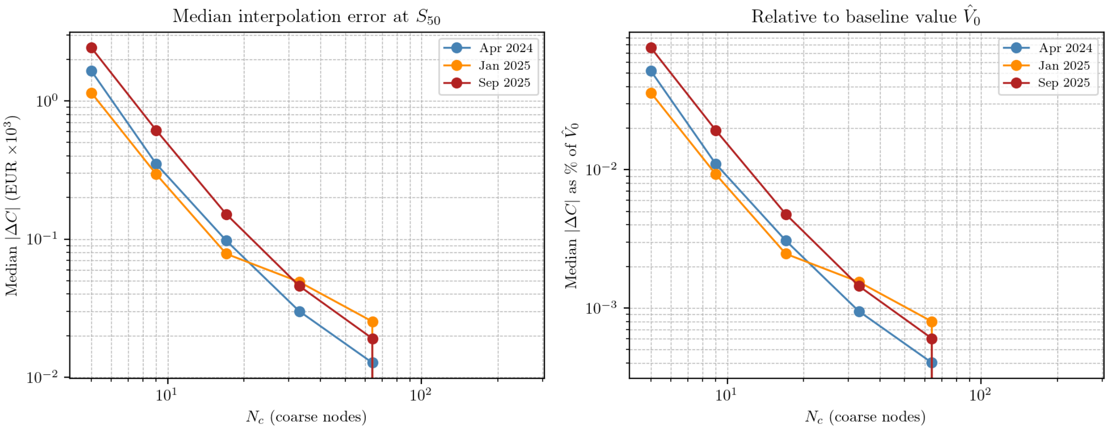


Figure 6.27: Median interpolation error in $C(t, v, S_{50})$ for the Dutch salt cavern contract, where S_{50} denotes the median spot price. Left: absolute error in EUR. Right: error relative to $\hat{V}_0$. Each curve corresponds to a representative date.

The near-linearity of C in v explains why interpolation itself contributes virtually no error, but it does not eliminate the effect of solving the dynamic programme on a coarse set of regression nodes. With too few coarse nodes, intermediate inventory levels are not represented explicitly in the regression step, and the resulting policy loses resolution. Because this effect accumulates over the full backward recursion, it can still generate a noticeable bias in $\hat{V}_0$ even when interpolation between adjacent nodes is close to exact.

On the basis of these results, we have adopted $\alpha_c = 10{,}000$ MWh, corresponding to $N_c = 51$, as the reference grid for the OLS simulation study. This choice provides a satisfactory compromise between accuracy and runtime and is sufficiently close to the fine-grid benchmark for the purposes of the simulation study.

Normality of the value distribution

According to the asymptotic analysis of Clément et al. (2002), the normalised error of the LSMC estimator is asymptotically Gaussian in the American option setting. This does not imply that the finite-sample distribution of the estimated values is itself exactly normal, but it does provide a useful benchmark for assessing whether the regression procedure generates severe distributional distortions. At the same time, Kircher (2004, as cited in Boogert and De Jong (2007)) reports that high-order polynomial regressions in Least Squares Monte Carlo may generate asymmetric fat tails in the resulting price distribution, particularly when more than four terms are included in the regression function. This makes it relevant to examine whether increasing the polynomial basis produces similar instability here.

Figure A.11 shows that no such pattern emerges in the present setup. Across all tested basis degrees, skewness remains close to zero and excess kurtosis stays slightly negative, so the finite-sample pathwise distribution of $\hat{V}_0$ remains approximately symmetric and, if anything, somewhat lighter-tailed than Gaussian. The Shapiro–Wilk and Jarque–Bera tests still reject exact normality for some specifications, but the deviations are small and do not become more pronounced as the basis degree increases. Thus, while this exercise does not test the asymptotic Gaussian-error result of Clément et al. (2002) directly, it indicates that the value estimates do not display the pronounced non-normality that Kircher associates with unstable high-order polynomial regressions.

6.8.3 Sensitivity analysis

In this section, we examine how the Dutch salt cavern valuation responds to changes in both market-model parameters and operational characteristics. Whereas the preceding subsection focussed on numerical robustness of the LSMC implementation, the purpose here is to assess how the estimated storage value changes when assumptions of the simplified contract are varied around the baseline specification.

Sensitivity to mean reversion and volatility

We first vary the two central parameters of the one-factor spot-price model, namely the mean-reversion rate κ and the volatility parameter σ. These parameters govern the persistence and amplitude of spot-price fluctuations and therefore directly affect the value of operational flexibility.

In the baseline specification, these parameters are estimated from the historical Dutch spot-price sample used in the simulation study, giving $\kappa = 0.0084$ when time is measured in days and $\sigma = 0.0235$ per $\sqrt{\text{day}}$. The implied mean reversion is therefore fairly gradual. After one day, a shock retains about $\exp(-0.0084) \approx 0.9916$, corresponding to 99.16% of its size, and the associated half-life for deviations of the log spot price from its mean is roughly 82 days. The volatility estimate reflects the diffusion scale of the log-price process and should accordingly be interpreted over horizons that grow with the square root of elapsed time, rather than as a per-day volatility rate. Since both estimates are specific to the sample period, the sensitivity analysis below examines how strongly the valuation changes when κ and σ are varied around these baseline values.

Figure 6.28 shows that the value is much more sensitive to volatility than to mean reversion in the present setup. The lower panel displays an almost linear increase in $\hat{V}_0$ as σ increases, which is economically intuitive. Higher spot-price volatility creates larger spreads between favourable injection and withdrawal opportunities and therefore increases the option value of storage. By contrast, the upper panel shows a hump-shaped relation with respect to κ. $\hat{V}_0$ is increasing in κ up to a local maximum and decreasing thereafter.

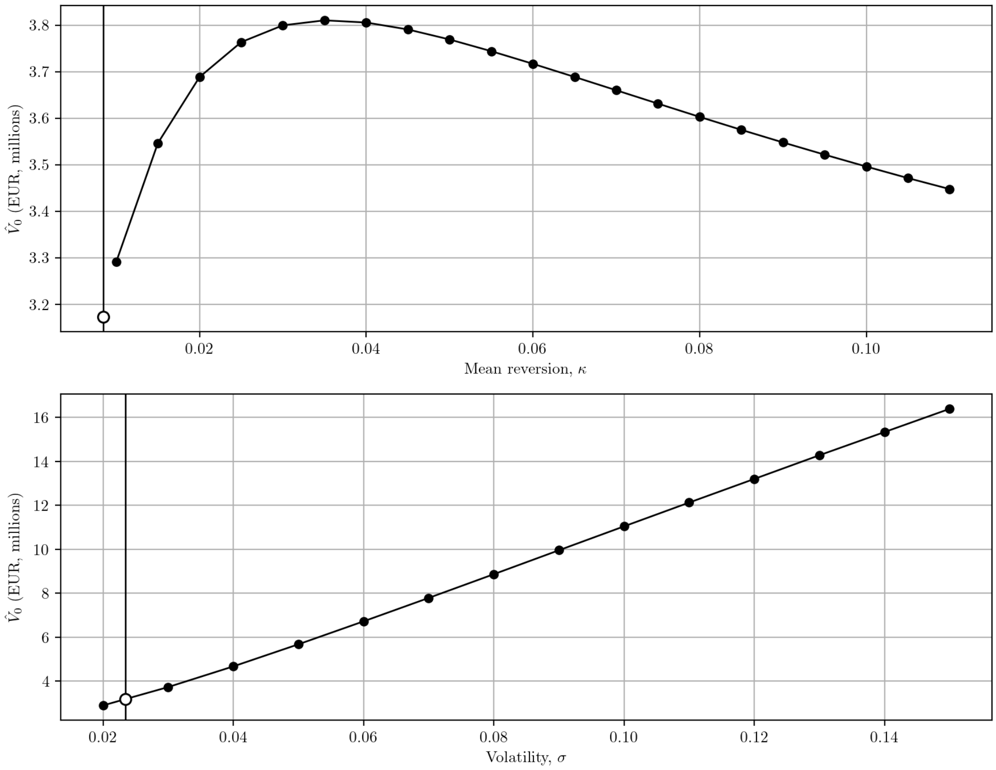


Figure 6.28: Sensitivity of the OLS-based storage value $\hat{V}_0$ to the mean-reversion rate κ and volatility σ in the Dutch salt cavern study. In each panel, the vertical line marks the baseline estimated parameter value from historical data.

This pattern is closely aligned with the interpretation given by Boogert and De Jong (2007, pp. 90–91). They argue that mean reversion affects storage value through two opposing channels. On the one hand, stronger mean reversion makes price movements more predictable and can improve the timing of injection and withdrawal decisions, thereby raising $\hat{V}_0$. On the other hand, it also pulls prices back more quickly towards their long-run level, reducing the size and duration of price excursions and thereby limiting arbitrage opportunities. Boogert and De Jong find that the resulting value profile is fairly stable over a reasonable range of κ, while volatility has the more pronounced impact. The same qualitative conclusion emerges in our analysis, where the estimated value changes only moderately across the considered range of mean-reversion rates, whereas the response to σ is much steeper.

Sensitivity to operational characteristics

We next consider the operational side of the contract by varying the withdrawal-rate cap, the injection-rate cap, and the working-gas capacity one at a time. This makes it possible to identify which contractual characteristic is most restrictive around the baseline specification and therefore has the largest marginal impact on value.

Figure 6.29 shows that the storage value increases monotonically when the operational constraints are relaxed, but with clear diminishing marginal gains. Raising the withdrawal-rate cap increases $\hat{V}_0$, though the curve is relatively flat once the withdrawal flexibility is already moderate. By contrast, the middle panel is steeper around the baseline specification, which indicates that the value is more sensitive to changes in the injection-rate cap in the present contract. The bottom panel shows that expanding the working-gas capacity likewise raises value monotonically across the entire range considered, with diminishing but still clearly positive marginal gains even at the largest volumes.

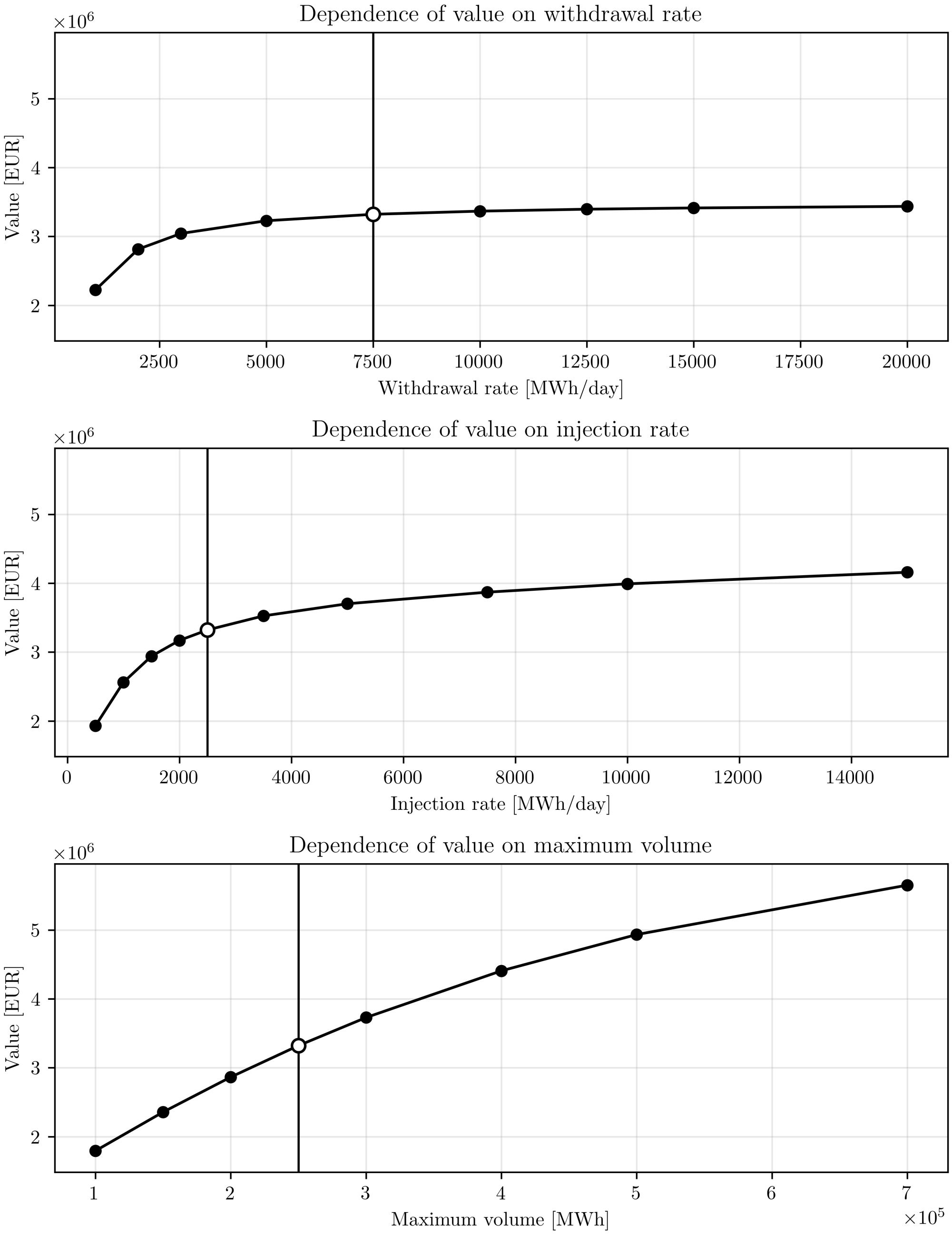


Figure 6.29: Sensitivity of the OLS-based storage value $\hat{V}_0$ to the withdrawal-rate cap, injection-rate cap, and working-gas capacity in the Dutch salt cavern study. In each panel, the vertical line marks the baseline contract specification.

This qualitative pattern is closely aligned with the sensitivity results reported by Boogert and De Jong (2007, pp. 91–92). They likewise find that storage value rises most strongly when injection, withdrawal, or capacity is increased from low levels, and that the effect flattens as the corresponding constraint becomes less binding. Their interpretation is that the economically most relevant expansion is the one that relaxes the currently tightest bottleneck. The same conclusion appears to hold here. In our baseline specification, the injection rate is lower relative to the working volume than the withdrawal rate, and the figure suggests that relaxing this margin yields the strongest payoff near the reference contract. Thus, while the quantitative sensitivities differ across specifications, the economic message is the same as in Boogert and De Jong, namely that marginal expansions are most valuable when they target the operational characteristic that is currently most restrictive.

6.9 Summary of valuation results

Tables 6.30 and 6.31 collect the estimated in-sample and out-of-sample storage values together with wall-clock runtimes for all six methods applied to the GSD and Dutch salt cavern cases, respectively. For the intrinsic and rolling intrinsic benchmarks, which are deterministic, an out-of-sample estimate is not applicable.

Across both cases, spline regression and OLS produce the closest value estimates to each other and offer the best accuracy-to-runtime ratio among the spot-based methods. Notably, all neural-network specifications produce estimates below the rolling intrinsic lower bound in both cases, meaning none of the neural-network methods surpass even this deterministic benchmark. The MLP with joint inventory input comes closest, but at a substantially higher computational cost of roughly 3.7 hours for the GSD contract and 28 minutes for the simpler Dutch contract. In the GSD case, the KAN specification with joint inventory input outperforms the nodewise variant but still falls short of the polynomial regression benchmarks in the current configuration. In the Dutch case, the KAN with joint inventory input actually performs worse than the nodewise variant. The MLP does not exhibit this reversal, suggesting that the behaviour is specific to the KAN architecture, but we leave a precise explanation for future work.

Table 6.30: Summary of valuation results for the GSD case. Values are in millions of euros (€M) rounded to three decimal places. Out-of-sample estimates are not applicable (—) for the deterministic benchmark methods. Runtimes are in H:MM:SS format.

Method	In-sample (€M)	Out-of-sample (€M)	Runtime
Intrinsic	78.999	—	0:00:01
Rolling Intrinsic	107.245	—	0:59:00
OLS	123.481	123.041	0:03:25
Spline	124.247	123.263	0:07:20
MLP	98.856	97.099	1:42:57
MLP (joint input)	107.101	106.727	3:42:20
KAN	88.619	86.649	34:47:43
KAN (joint input)	94.081	93.503	0:32:59

Table 6.31: Summary of valuation results for the Dutch salt cavern case. Values are in millions of euros (€M) rounded to three decimal places. Out-of-sample estimates are not applicable (—) for the deterministic benchmark methods. Runtimes are in H:MM:SS format.

Method	In-sample (€M)	Out-of-sample (€M)	Runtime
Intrinsic	2.179	—	0:00:01
Rolling Intrinsic	3.258	—	0:00:08
OLS	3.321	3.317	0:00:06
Spline	3.343	3.340	0:00:06
MLP	2.824	2.800	0:19:01
MLP (joint input)	2.963	2.961	0:28:16
KAN	2.738	2.720	5:27:48
KAN (joint input)	2.321	2.318	0:19:59

We also explored alternative hyperparameter configurations for the joint-input variants of both neural-network methods on the Dutch salt cavern contract. For the KAN with joint input, increasing width and depth, lowering the learning rate, enabling grid extension during training, and increasing the number of simulated paths while holding the per-step batch size fixed all produced lower estimates, while applying regularisation and pruning (see Chapter 4) produced a slightly higher estimate. Only scaling the total path count yielded a substantial increase, at the cost of considerably longer computation times, pointing to training data volume as the primary bottleneck for the KAN.

For the MLP with joint input, wider or deeper architectures, more epochs, and a larger L-BFGS history size each gave incremental gains. The dominant improvement, however, came from a single architectural change: replacing ReLU with SiLU as the activation function. This alone raised the in-sample estimate from €2.963 million to €3.301 million, effectively closing the gap with the regression benchmarks and placing the MLP within 0.6% of the OLS estimate of €3.321 million and 1.3% of the spline estimate of €3.343 million. SiLU is everywhere differentiable and avoids the dead-neuron problem of ReLU.

7 | Discussion

In this chapter, we discuss the methods, results, and limitations of the models developed in this thesis. We also examine the implications of the findings for commodity storage valuation and outline potential directions for future research.

7.1 Performance of approximation methods

7.1.1 Why simple models outperform complex models

The central empirical finding of this thesis is that polynomial OLS regression and spline regression outperform both MLP- and KAN-based approximations in the LSMC valuation framework, despite neural networks being universal function approximators (Hornik et al., 1989). Understanding why requires examining the structure of the continuation value function that each method is asked to learn.

The grid convergence analysis in Subsection 6.4.2 and the joint-input experiments in Subsections 6.6.1 and 6.7.1 jointly reveal that the continuation value $C(t, v, S)$ is close to affine in the inventory level v across the feasible range, contrary to what Boogert and De Jong (2007) found. This is particularly evident in the Dutch salt cavern case, where the cross-sections of $C(t, \cdot, S_{50})$ are essentially linear in v. In the GSD case, the continuation value is somewhat more complex due to inventory-dependent injection and withdrawal rate constraints, but the degree of nonlinearity is modest. Polynomial regression of degree three is well-suited to such functions, and the gain from additional expressive capacity is minimal. The choice is also numerically safe, since the monomial design matrix becomes increasingly ill-conditioned as the degree rises, inflating coefficient variance without reducing approximation bias, an effect visible in Figure 6.21 at degree ten but not at degree three.

This structural observation explains why the nodewise MLP underperforms relative to OLS by approximately 20% in the GSD case. With up to $T \times N_c \approx 729 \times 254 \approx 185{,}000$ separate networks trained during the backward pass, each network approximates a univariate function of the spot price at a single fixed inventory level. Although the full set of M simulated paths contributes to each nodewise regression, the network cannot share information about how the continuation value varies across inventory levels. This is precisely the information that the near-affine structure would allow a shared model to exploit efficiently. The nodewise design is therefore costly relative to its expressiveness, training many small models that each solve an unnecessarily restricted approximation problem, when a simpler model that treats inventory as an additional input would make better use of the available data.

7.1.2 Effects of the joint-input architecture

The near-affine structure of the continuation value directly motivates the joint-input architecture tested in Subsections 6.6.1 and 6.7.1, in which inventory is passed as an explicit input to a single shared network rather than handled through separate nodewise models. For the MLP, the improvement is consistent across both cases. The joint-input MLP raises the estimated GSD storage value by 8.34% relative to the nodewise baseline (from €98.856M to €107.101M) and similarly improves the Dutch estimate. For the KAN, the picture is mixed. The joint-input KAN raises the GSD estimate by 6.1% (from €88.619M to €94.081M), bringing it closer to the OLS benchmark, but reduces the Dutch estimate by 15.23% relative to the nodewise KAN. As noted in Section 6.9, the MLP does not exhibit this reversal, suggesting the behaviour is specific to the KAN architecture, though the precise mechanism is left as a direction for future research (see Section 7.4).

In both the MLP and KAN cases for the GSD problem, a residual gap to OLS remains. The joint-input MLP and KAN fall approximately 13% and 24% short of the OLS benchmark, respectively, indicating that the shared-network formulation does not fully overcome the limitations of the neural network approach in the present configuration. The hyperparameter analysis reported in Section 6.9 nonetheless shows that replacing ReLU with SiLU as the MLP activation function alone closes this accuracy gap almost entirely. The performance disadvantage of the MLP relative to OLS is therefore partly a configuration-specific outcome rather than a purely structural property of the approximation problem. The case against neural network methods in the present setting therefore rests primarily on computational cost.

7.1.3 Computational cost and practical viability

Beyond approximation quality, the computational cost of the neural network approaches is a serious practical limitation in the present setting, and it is not resolved by configuration choices that close the accuracy gap. The nodewise KAN requires approximately 34 hours and 48 minutes of wall-clock time for the GSD case, compared to 3 minutes and 25 seconds for OLS, which is a factor of roughly 600. The joint-input variants reduce this to 33 minutes (KAN) and 3 hours 42 minutes (MLP), yet both still underperform OLS on accuracy. Spline regression, by contrast, achieves the highest in-sample estimate (€124.247M versus €123.481M for OLS) in only 7 minutes and 20 seconds.

7.1.4 Structural limitations of neural network methods

Two practical factors limit both MLP- and KAN-based approximations relative to polynomial regression in this setting. After the backward recursion conditions on time, each network approximates a function of at most two variables. The expressive capacity of neural networks over direct polynomial representations is most relevant when the input dimension is large enough that the latter becomes inefficient. At dimension one or two, a degree-three polynomial fitted by least squares is already an accurate and efficient approximation, and each network must converge to a qualitatively similar function through iterative training instead. The continuation value is also empirically near-linear in both state variables, leaving little scope for the nonlinear activations in either architecture to offer an advantage over OLS within the limited iteration budgets used here.

Additional concerns apply specifically to KANs. The Kolmogorov–Arnold representation theorem guarantees that any continuous multivariate function can be expressed as a finite composition of continuous univariate functions, which has been interpreted as motivating KANs as natural approximators. The inner functions $\phi_{q,p}$ in the Kolmogorov–Arnold representation need not be smooth, making them infeasible to learn in practice. Relatedly, Vitushkin (1954) proved that there exist r-times continuously differentiable functions of $n \geq 2$ variables not representable by superposition of r-times continuously differentiable functions of fewer than n variables, for $r = 1, 2, \ldots$. Girosi and Poggio (1989) reinforce this conclusion, arguing that the $\phi_{q,p}$ functions cannot be practically approximated and that Φ_q is at least as complex as the original high-dimensional function.

Liu et al. (2025) argue that most functions encountered in science and engineering are smooth and have sparse compositional structure, making typical cases more tractable than the worst-case bound suggests. This justification, however, does not apply to the continuation value functions arising in dynamic programming. The near-affine behaviour documented in Subsection 6.4.2 characterises the continuation value within the feasible interior, where state constraints are not binding. The function as a whole is not smooth: infeasible states carry a value of $-\infty$ and the Bellman optimisation introduces kinks at the bang-bang decision boundaries, making it discontinuous and non-differentiable in ways that are fundamental to the optimal control structure rather than incidental. This suggests a potential mismatch between KAN's design assumptions and the properties of the target function class in this particular setting. The spline activations in the present KAN implementation also use a grid of three intervals placed uniformly over the training data range, concentrating capacity in the centre of the spot-price distribution and leaving the tails poorly covered. Polynomial basis functions are globally defined and not subject to this constraint, and the adaptive grid extension available in KAN requires sufficient training iterations to identify and correct misaligned regions, which is not achievable within the five-step budget used here.

7.2 Economic interpretation of results

7.2.1 Extrinsic value and the limits of forward-based strategies

It is important to distinguish between the theoretical value of the optimal spot-based strategy and the value that can be realised in practice. The rolling intrinsic strategy is directly executable, as it requires only the forward curve, which is continuously observable, and its trades are straightforwardly implemented in the forward market. The LSMC valuation, by contrast, prescribes a policy that depends on the realised spot price at each date, which requires active spot market participation. This introduces execution risk, as actual trades occur at bid or ask prices rather than mid-market closing prices, and the realised outcome will depend on the accuracy of the price model. The best spot-based LSMC estimate (€124.247M for the spline method) should therefore be interpreted as a theoretical upper bound on the achievable spot-based value, and the realised premium over the rolling intrinsic strategy will generally be lower.

7.2.2 The economic cost of regulatory filling requirements

The sensitivity analysis reported in Table 6.10 allows the EU filling requirements (European Commission, 2023, 2024) to be evaluated in economic terms. Removing all intermediate constraints raises the estimated OLS value from €123.481M to €163.890M, an increase of 32.73%, indicating that the regulatory framework imposes a material cost by forcing injection strategies that do not always align with profit-maximising behaviour. Among the individual requirements, the February winter-floor proves most binding (3.06% gain when relaxed from 45% to 20%), while the November 90% obligation and the 2025 September easing contribute only modestly; see Table 6.10 for details. This is consistent with actual Danish storage behaviour, where the 90% November target was missed in both 2024 and 2025 (Danish Energy Agency, 2025), reflecting market conditions in which full injection was not incentivised.

7.2.3 Volatility as the dominant driver of storage value

The sensitivity analysis of the Dutch salt cavern contract reported in Figure 6.28 shows that storage value is substantially more sensitive to spot price volatility σ than to the mean-reversion rate κ. The estimated value increases approximately linearly in σ, since higher volatility widens the spreads between favourable injection and withdrawal opportunities.

The relationship with κ is hump-shaped, where moderate mean reversion improves trade timing, while stronger reversion suppresses price swings and limits arbitrage. Both findings are closely consistent with Boogert and De Jong (2007). An implication for operators is that changes in market volatility, driven by supply disruptions or shifts in renewable penetration, should be expected to have a larger impact on storage value than changes in the speed of price reversion.

7.2.4 Bang-bang nature of optimal policies

Across all six valuation methods, the realised inventory decisions are concentrated at the feasible extremes of maximum injection, maximum withdrawal, or inactivity. This bang-bang property is a structural feature of the optimal control problem rather than a numerical artefact. Because the injection and withdrawal costs are linear in quantity, with the payoff at each step proportional to the amount traded at the prevailing price, the Bellman equation always selects corner solutions whenever a spread is strictly profitable. Intermediate trades are only observed when operational constraints (such as the inventory-dependent injection and withdrawal rate limits) reduce the maximum admissible action below the unconstrained optimum, or when no profitable spread exists. The practical implication is that gas storage operations naturally take outright positions, and the primary source of value is the timing of trades rather than any form of partial utilisation.

This structure has a direct mathematical explanation. The payoff $G(S(t), \Delta v)$ is linear in Δv by construction. When the continuation value C is affine in the post-action inventory $v(t) + \Delta v$, as found empirically in both cases studied, the continuation term is also affine in Δv, making the total objective affine. Maximising an affine function over the compact set $\mathcal{D}(t, v(t))$ always yields a boundary solution, so the optimal Δv is either maximum injection, maximum withdrawal, or inactivity. If the continuation value were strictly concave in inventory, the total objective would be strictly concave in Δv and an interior optimum would be possible. The observed bang-bang behaviour is therefore a consequence of the empirically near-linear inventory dependence of the continuation value, and need not arise in storage problems where the continuation value is strictly concave.

7.3 Model and methodological limitations

7.3.1 The one-factor spot price model

Throughout this thesis, spot price dynamics are modelled using the one-factor mean-reverting model of Schwartz (1997), in which the log-price follows an Ornstein–Uhlenbeck process with a time-varying deterministic mean. The model has, however, significant limitations. The one-factor structure cannot accommodate volatility clustering, sudden price spikes, or regime shifts, all of which have been prominent features of European gas markets. The energy crisis of 2021–2022, during which TTF prices reached unprecedented levels before reverting sharply, represents precisely the kind of extreme behaviour that the Schwartz model cannot replicate. Since the model is calibrated to a relatively calm period, the estimated parameters do not reflect the tail behaviour observable in historical data, and the resulting valuation estimates should not be extrapolated to conditions of market stress.

A formal diagnostic of the estimated model supports this concern. The AR(1) residuals from the joint estimation of the log-price process are examined in Figure 7.1. Both the Jarque–Bera test and the Shapiro–Wilk test reject the null hypothesis of normally distributed innovations at any conventional significance level. The estimated excess kurtosis of 2.33 indicates that large innovations occur far more frequently than the Gaussian OU model implies, while the mild negative skewness (−0.20) reflects a slight asymmetry in the residual distribution. The Q–Q plot in Figure 7.1 confirms this, where points in both tails deviate significantly from the normal reference line, whereas the centre of the distribution is well described by the fitted normal. These departures are consistent with a leptokurtic innovation distribution and indicate that the model systematically underestimates the probability of extreme price moves.

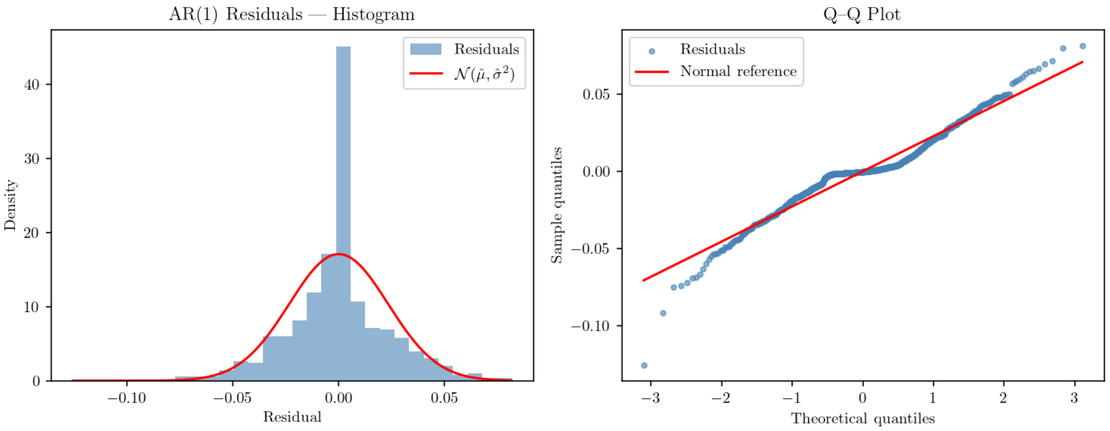


Figure 7.1: Normality diagnostics for the AR(1) residuals from the joint estimation of the log-price model. Left: histogram of residuals with a fitted normal density overlaid. Right: normal Q–Q plot with a reference line. Both panels indicate fat-tailed departures from normality, with an excess kurtosis of 2.33 and Jarque–Bera $p < 0.001$.

7.3.2 Transaction costs and execution risk

The valuation framework assumes frictionless trading, in which injection and withdrawal decisions are executed at the prevailing spot or forward price without transaction costs, bid-ask spreads, or market impact. In practice, each trade incurs costs that reduce the realised payoff relative to the theoretical value. This is particularly relevant for the rolling intrinsic strategy, which requires re-optimising and re-trading every day throughout the two-year action period. The cumulative transaction costs from frequent re-trading could materially reduce the realised rolling intrinsic value below the theoretical €107.245M figure. For the spot-based LSMC approach, the same concern applies, compounded by the additional uncertainty of executing at realised spot prices rather than observed forward quotes.

A related issue is the use of mid-market closing prices in the construction of the forward curve. As noted in Subsection 1.2.1, relying on closing prices rather than firm bid and ask quotes may overstate intrinsic value, since actual trades must be executed at prices that include the bid-ask spread.

7.4 Future research

Several natural extensions follow from the limitations and findings discussed above. The most immediate is the adoption of a richer spot price model. Incorporating a second stochastic factor or a jump component would add realism and would also raise the dimensionality of the state space. The one-dimensional state space of the present study, where price is the sole stochastic driver, may have constrained the ability of MLP and KAN to demonstrate a comparative advantage over OLS, and a higher-dimensional state space may therefore improve their performance relative to OLS.

On the architectural side, the results of this thesis suggest that the nodewise training design is a primary source of the gap between MLP/KAN and OLS. The joint-input MLP consistently outperforms its nodewise counterpart across both cases studied, suggesting that shared-network architectures merit further investigation. The evidence for the KAN is more mixed, as the joint-input variant improves over the nodewise design for GSD but not for the Dutch contract, and understanding when and why the shared architecture helps or hurts would be a useful direction for future work.

8 | Conclusion

This thesis has examined the valuation of natural gas storage contracts using the least squares Monte Carlo framework, with a focus on comparing approximation methods for the continuation value function. Six approximation methods were evaluated within the LSMC backward recursion: polynomial OLS regression, spline regression, a nodewise MLP, a joint-input MLP, a nodewise KAN, and a joint-input KAN. Two contracts were studied. The first was a real-world Danish storage system operated by Gas Storage Denmark (GSD), comprising the Stenlille aquifer and the Lille Torup salt cavern facility and subject to EU seasonal filling requirements, and a stylised Dutch salt cavern contract used for convergence and sensitivity analysis.

The theoretical foundation was established through a discrete-time stochastic optimal control formulation, in which the storage operator maximises expected discounted profits subject to inventory and operational constraints. Spot price dynamics were modelled using the one-factor mean-reverting process of Schwartz (1997), calibrated to a forward curve constructed from Trading Hub Europe (THE) closing prices. The LSMC algorithm of Longstaff and Schwartz (2001) was adapted to this setting by replacing the standard regression step with each of the six approximation methods. Theoretical properties of each method were analysed prior to the empirical study, including the representational capacity of KANs via the Kolmogorov–Arnold theorem and the numerical conditioning of polynomial regression at varying degrees.

The main empirical finding is that polynomial OLS regression and spline regression outperform both MLP- and KAN-based methods on every dimension examined. For the GSD contract, the OLS estimate of €123.481M and the spline estimate of €124.247M exceed the best neural network estimates, with the joint-input MLP at €107.101M (+8.34% over its nodewise baseline) and joint-input KAN at €94.081M (+6.1% over its baseline), leaving residual gaps of approximately 13% and 24% relative to OLS, respectively. The gap is explained by the empirically near-affine dependence of the continuation value on inventory, which makes polynomial regression well-suited to the target function while limiting the benefit of additional neural network capacity. Computationally, the nodewise KAN requires approximately 34 hours and 48 minutes compared to 3 minutes and 25 seconds for OLS, a factor of roughly 600. The joint-input architecture reduces this overhead but does not close the accuracy gap. The bang-bang structure of optimal policies (maximum injection, maximum withdrawal, or inactivity) was observed consistently across all methods and is a direct consequence of the affine objective induced by the near-linear continuation value. On the GSD contract, the LSMC strategy yields an estimated extrinsic value of approximately €17.0M over the rolling intrinsic benchmark of €107.245M. Removing all EU seasonal filling requirements raises the OLS estimate by 32.73% (from €123.481M to €163.890M). A sensitivity analysis of the Dutch salt cavern confirms that storage value is substantially more sensitive to spot price volatility than to the mean-reversion rate, consistent with the findings of Boogert and De Jong (2007).

The analysis is subject to several limitations. The one-factor Schwartz model cannot replicate volatility clustering, price spikes, or regime shifts, and formal diagnostics confirm fat-tailed innovations with excess kurtosis of 2.33 and rejection of normality. The valuation framework assumes frictionless trading, and the theoretical LSMC values should be interpreted as upper bounds on achievable profits. Neural network methods also face structural limitations in the present setting. After conditioning on time, each network approximates a function of at most two state variables, and degree-three polynomial regression is already an accurate and efficient approximation at this dimension, leaving little scope for additional expressive capacity.

Overall, the results of this thesis indicate that, for the class of gas storage problems studied, well-tuned polynomial regression is both more accurate and orders of magnitude faster than neural network alternatives. The near-affine continuation value simultaneously explains the superiority of OLS, the bang-bang optimal policies, and the limited gains from neural network expressiveness. Neural network methods are most likely to offer a meaningful advantage in higher-dimensional state spaces, for instance when richer price processes introduce additional stochastic factors, a direction that represents the most natural extension of this work.

References

Becker, S., Cheridito, P., and Jentzen, A. (2020). Deep optimal stopping. arXiv:1804.05394 [math.NA].

Benth, F., Koekebakker, S., and Ollmar, F. (2007). Extracting and applying smooth forward curves from average-based commodity contracts with seasonal variation. *The Journal of Derivatives*, 15(1):52–66.

Boogert, A. and De Jong, C. (2007). Gas Storage Valuation Using a Monte Carlo Method. *The Journal of Derivatives*, 15(3):81–98.

Breslin, J., Clewlow, L., Elbert, T., Kwok, C., and Strickland, C. (2008). Gas storage: Overview & static valuation. Technical report, Lacima Group.

Bundesnetzagentur (2026). Gas Day-Ahead Prices – Trading Hub Europe. Accessed: 2026-04-16.

Clément, E., Lamberton, D., and Protter, P. (2002). An analysis of a least squares regression method for American option pricing. *Finance and Stochastics*, 6(4):449–471.

Danish Energy Agency (2025). Filling of gas storage for winter 2025–2026. Accessed: 2026-03-23.

de Boor, C. (1978). *A Practical Guide to Splines*, volume 27 of *Applied Mathematical Sciences*. Springer.

Delbaen, F. and Schachermayer, W. (1994). A general version of the fundamental theorem of asset pricing. *Mathematische Annalen*, 300(1):463–520.

Esary, J. D., Proschan, F., and Walkup, D. W. (1967). Association of random variables, with applications. *The Annals of Mathematical Statistics*, 38(5):1466–1474.

European Commission (2023). Commission Implementing Regulation (EU) 2023/2633 of 20 November 2023 setting the filling trajectory with intermediate targets for 2024 for each Member State with underground gas storage facilities on its territory and directly interconnected to its market area. Official Journal of the European Union. Accessed: 2026-03-23.

European Commission (2024). Commission Implementing Regulation (EU) 2024/2995 of 29 November 2024 setting the filling trajectory with intermediary targets for 2025 for each Member State with underground gas storage facilities on its territory and directly interconnected to its market area. Official Journal of the European Union. Accessed: 2026-03-23.

Fraunhofer ISE (2026). Energy Charts. Accessed: 2026-03-23.

Gas Storage Denmark (2026a). Operational data. Accessed: 2026-03-23.

Gas Storage Denmark (2026b). Our storage. Accessed: 2026-03-23.

Gas Storage Denmark (2026c). REMIT / reduced withdrawal capacity, ROY23. Accessed: 2026-03-23.

Girosi, F. and Poggio, T. (1989). Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant. *Neural Computation*, 1(4):465–469.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering.* Springer, 1st edition.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, NY, 2 edition. Corrected 12th printing, 13 January 2017.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114(5):953–956.

Lim, K. G. and Xiao, Q. (2002). Computing maximum smoothness forward rate curves. *Statistics and Computing*, 12(3):275–279.

Lin, H. W., Tegmark, M., and Rolnick, D. (2017). Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., and Tegmark, M. (2025). KAN: Kolmogorov–Arnold Networks. arXiv:2404.19756 [cs.LG].

Longstaff, F. and Schwartz, E. (2001). Valuing American Options by Simulation: A Simple Least-Squares Approach. *Review of Financial Studies*, 14(1):113–147.

Maragos, S. (2002). Valuation of the Operational Flexibility of Natural Gas Storage Reservoirs. In Ronn, E., editor, *Real Options and Energy Management.* Risk Books, London, UK.

Michaud, E. J., Liu, Z., and Tegmark, M. (2023). Precision machine learning. *Entropy*, 25(1):175.

Schwartz, E. and Smith, J. (2000). Short-term variations and long-term dynamics in commodity prices. *Management Science*, 46(7):893–911.

Schwartz, E. S. (1997). The stochastic behavior of commodity prices: Implications for valuation and hedging. *The Journal of Finance*, 52(3):923–973.

Sharma, U. and Kaplan, J. (2020). A neural scaling law from the dimension of the data manifold. arXiv:2004.10802 [cs.LG].

Vitushkin, A. G. (1954). On Hilbert's thirteenth problem. *Doklady Akademii Nauk SSSR*, 95:701–704.

A | Appendix

A.1 Activation functions

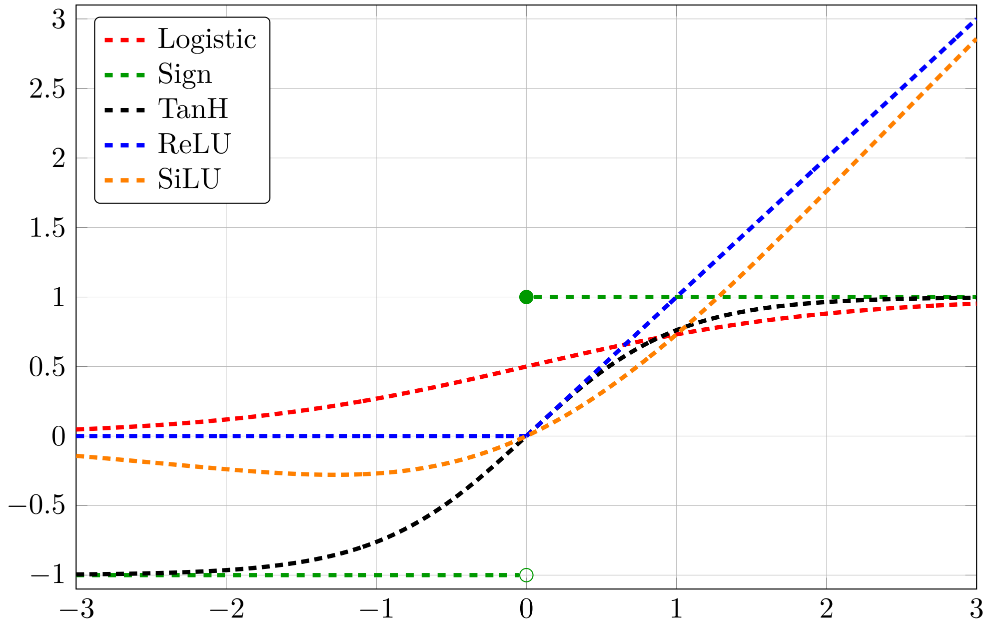


Figure A.1: Common activation functions used in neural networks. The logistic and hyperbolic tangent functions are smooth and bounded, the sign function is discontinuous at zero, the ReLU function is piecewise linear with a non-differentiable point at the origin, and SiLU is a smooth gating function defined as $x/(1 + e^{-x})$.

A.2 Regularisation

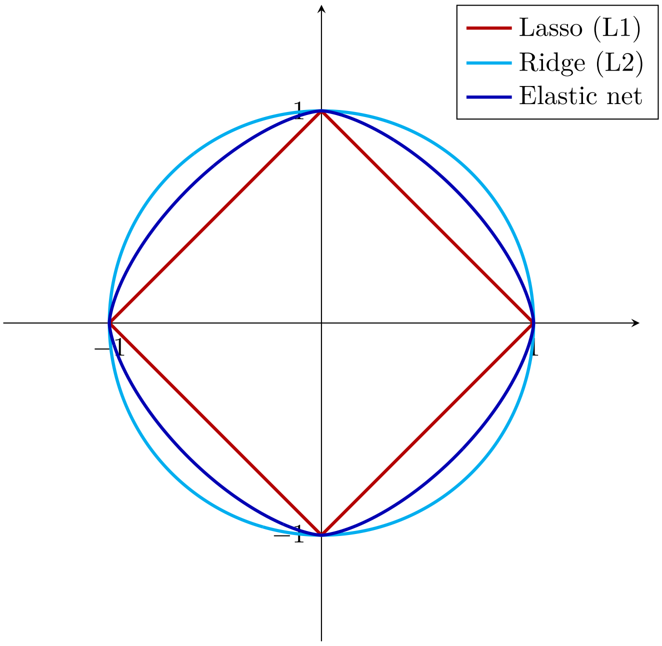
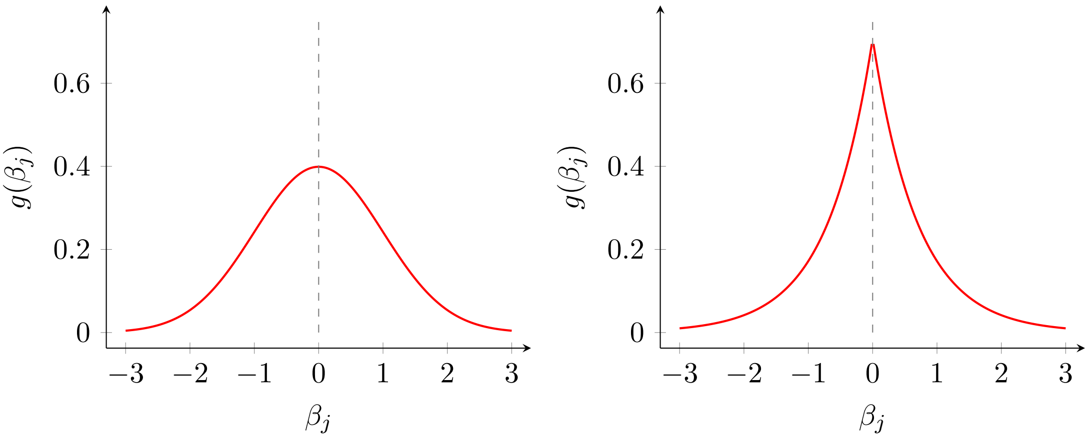


Figure A.2: Unit balls for different regularisation techniques in two dimensions. The Lasso (L1) regularisation promotes sparsity due to the corners of its unit ball, while Ridge (L2) regularisation tends to shrink coefficients uniformly. Elastic Net combines both approaches, balancing sparsity and coefficient shrinkage.



(a) Gaussian prior (Ridge)

(b) Laplace prior (Lasso)

Figure A.3: Priors corresponding to common regularisation techniques.

A.3 Gas Storage Denmark operational constraints

A.3.1 Injection and withdrawal rates

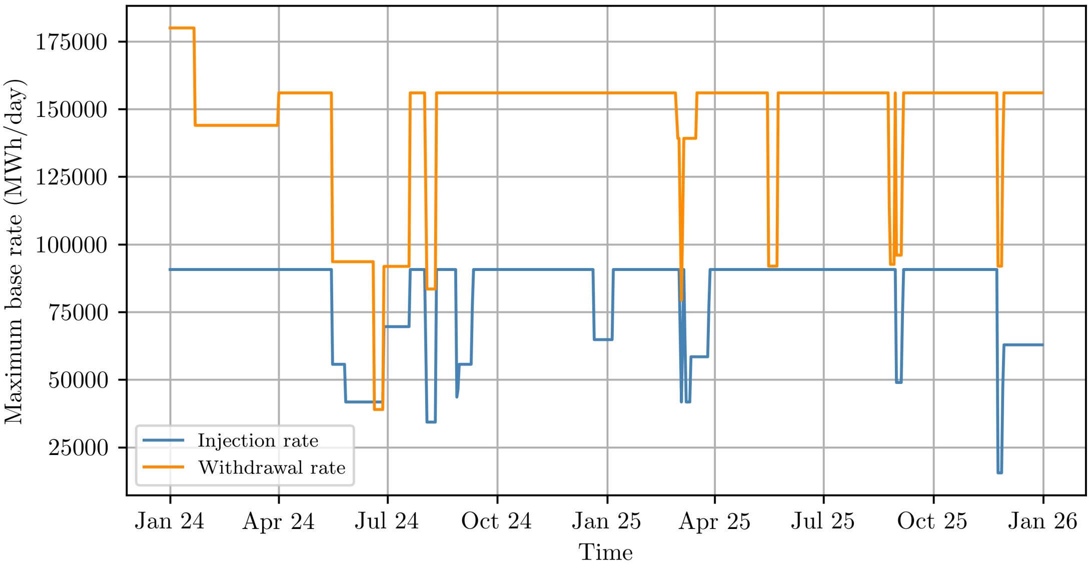


Figure A.4: Gas Storage Denmark's base injection and withdrawal rate constraints over time.

A.3.2 Maximum storage capacity

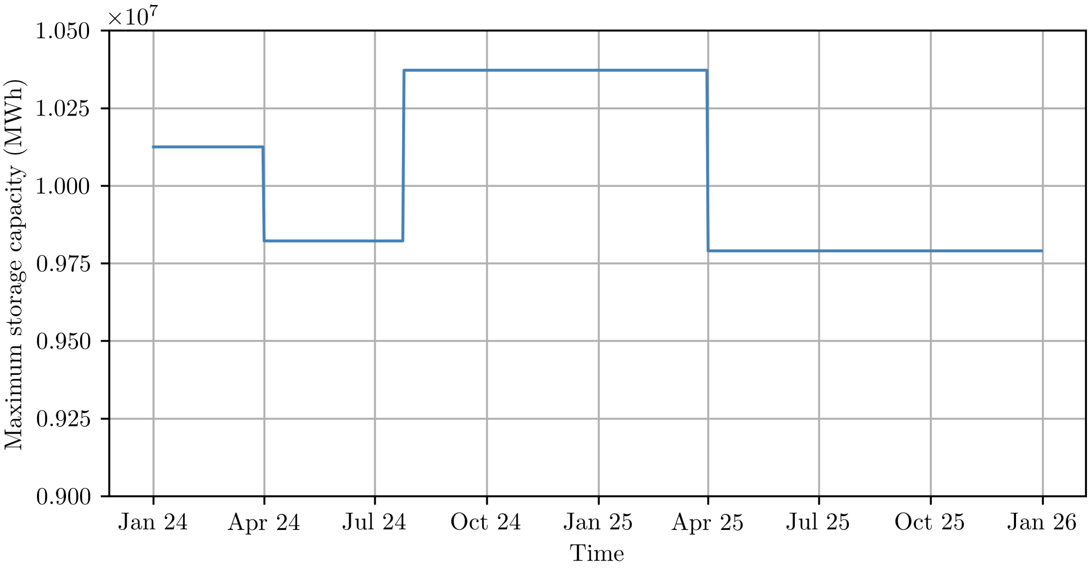


Figure A.5: Gas Storage Denmark's maximum storage capacity over time.

A.4 Valuation results

A.4.1 Rolling intrinsic valuation

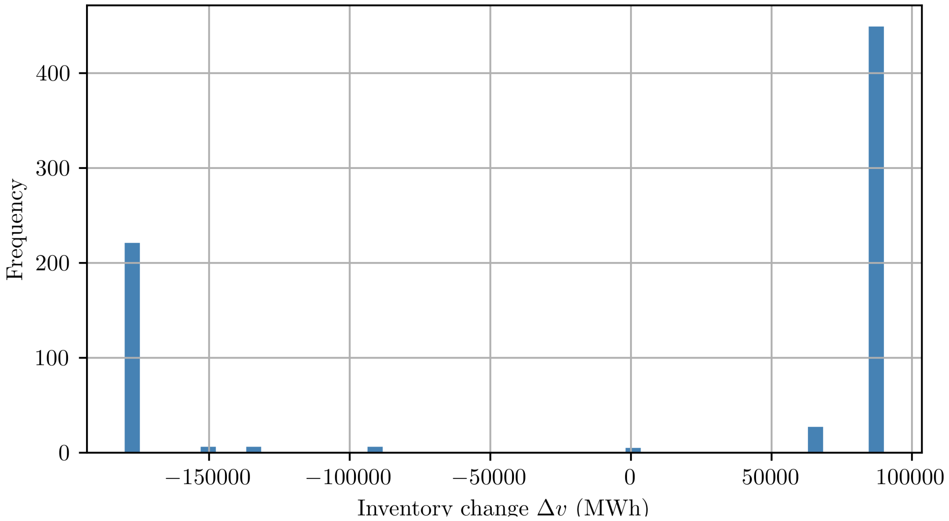


Figure A.6: Distribution of daily inventory changes across simulations using the rolling intrinsic valuation method.

A.4.2 OLS-based LSMC valuation

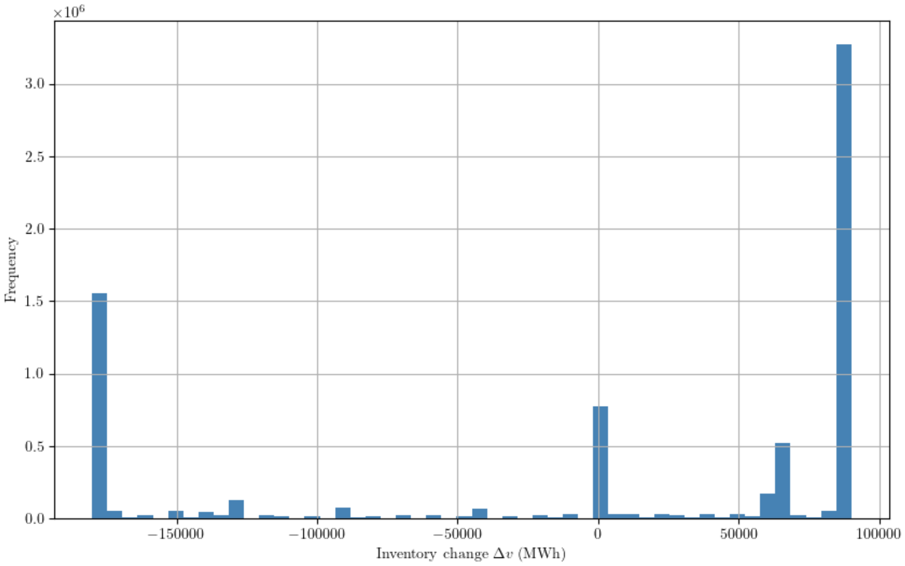


Figure A.7: Distribution of inventory changes across simulations using the OLS-based LSMC method.

A.4.3 Spline-based LSMC valuation

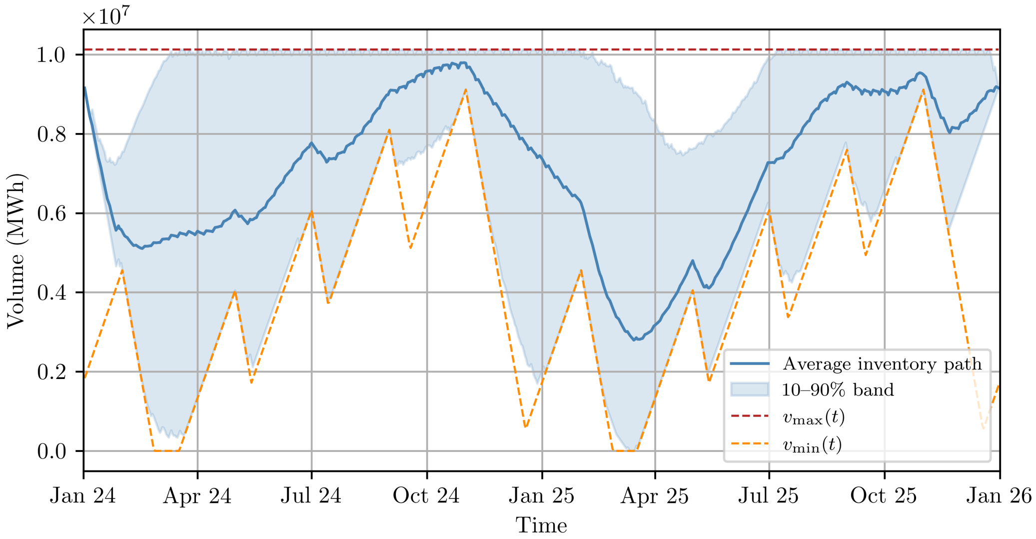


Figure A.8: Realised inventory paths across paths using the spline-based LSMC method. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the initial inventory level and the terminal target inventory level.

A.4.4 MLP-based LSMC valuation

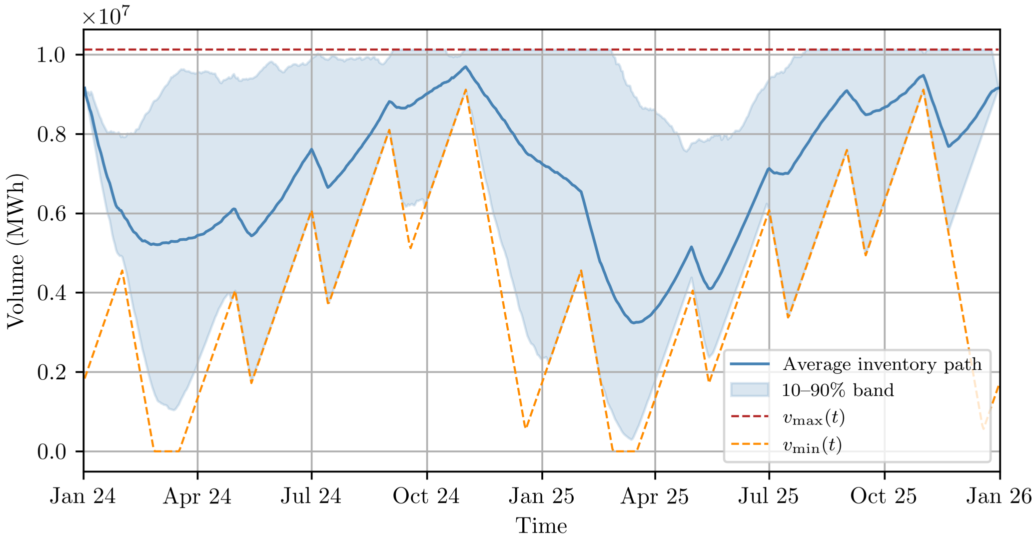


Figure A.9: Realised inventory paths across paths using the shared-input MLP-based LSMC method, in which the continuation model is trained jointly over price and inventory rather than separately at each inventory node. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the minimum and maximum inventory constraints.

A.4.5 KAN-based LSMC valuation

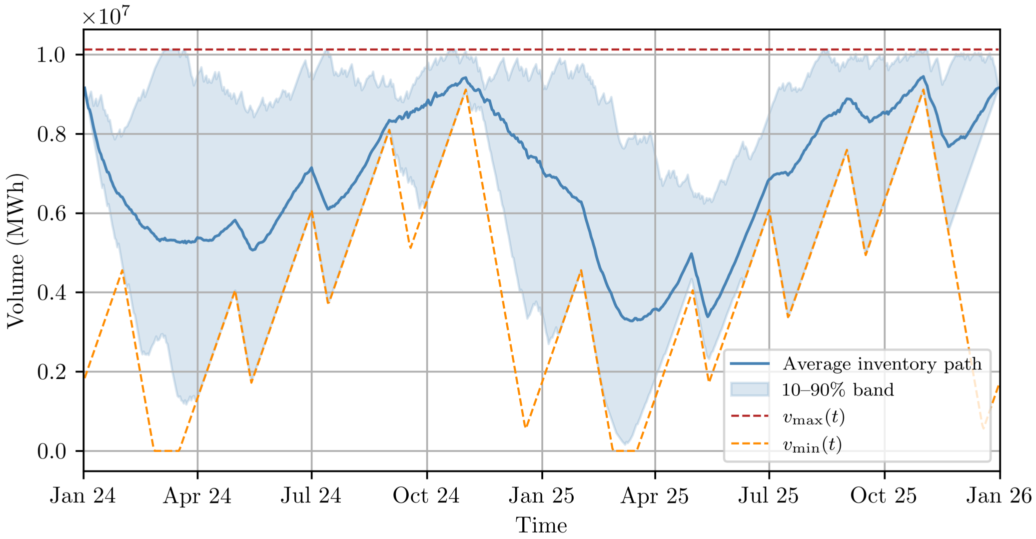


Figure A.10: Realised inventory paths across paths using the KAN-based LSMC method. The plot shows the average inventory path (solid line) and the 10–90% band (shaded area) across simulations. The dashed lines indicate the minimum and maximum inventory constraints.

A.5 Simulation study results

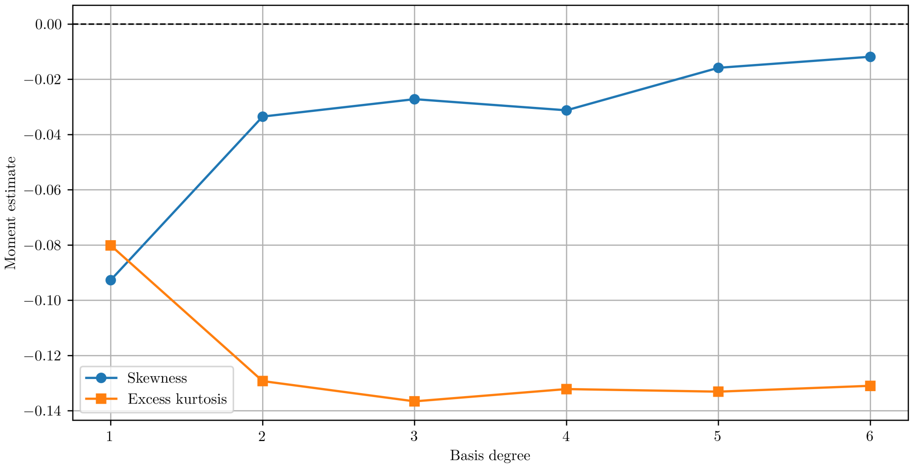


Figure A.11: Pathwise moments of the OLS valuation by basis degree.