Multi-Factor Least Squares Monte Carlo Gas Storage Valuation

Ubaid Hoque

May 2024

Contents

1 Introduction

In this piece of work, we explore the usage of the cmdty-storage python library to build various models that predict the behaviour of a gas storage facility over a 2 year period. First, we introduce a brief overview of the models used, the motivations for their use and an overview of the cmdty-storage library. Second, we introduce two price models which are then used in Monte Carlo simulations to model the behaviour of the gas spot price over the period concerned. Third, we build a gas storage valuation model, look at the valuation results and then the sensitivity of these results to the price models used. Finally, we outline how the usage of aforementioned models could be applicable to a fundamental gas trading model.

1.1 Least Squares Monte Carlo

Before diving into the gas storage valuation model itself, we assume the reader has a graduate level knowledge of statistics, with some introduction to option pricing. We can then introduce the Monte Carlo simulation method in the context of the valuation of an American Option for which there is no closed solution for, unlike the European case, and then extend this to gas storage valuation.

The valuation of gas storage facilities can be approached similarly to the valuation of American options using Monte Carlo simulation techniques. Consider a simplified example where the option's value is determined based on multiple paths of potential future prices at discrete time points.

Let $V_t(S)$ represent the value of the option at time t given the underlying asset price S. The valuation is performed backward, starting from the known final values at maturity, and proceeding to earlier times by considering both the option's intrinsic value and its expected future value under optimal exercise strategy.

1.1.1 American Option Valuation Example

To build an intuitive understanding, we first look at the proposed example in [5]. Consider an American put option, which can be exercised at discrete time points with a strike price of \$1.10 and expiry at time $t = 3$. Suppose we have simulated several possible paths for the underlying asset price at times $t = 1, 2, 3$ as shown in the table:

Path	**t=1**	**t=2**	**t=3**
1	1.09	1.08	1.34
2	1.16	1.26	1.54
3	1.22	1.07	1.03
4	0.93	0.97	0.92
5	1.11	1.56	1.52
6	0.76	0.77	0.90
7	0.92	0.84	1.01
8	0.88	1.22	1.34

Table 1: Table of values

At each time step t, starting from $t = 2$ (the option value in $t = 3$ is just the intrinsic one), the value of the option is calculated as:

$$V_t(S) = \max(\text{Intrinsic value}, \mathbb{E}[\text{Continuation Value}|\mathcal{F}_t]), \tag{1}$$

where $\mathcal{F}_t$ is the filtration representing information up to time t. The continuation value is the expected value of holding the option, computed using the risk-neutral probabilities and discount factors. Here we can consider the discount factor as simply the risk-free rate (6%).

At time $t = 3$, our payoff is clear. It is $\max(\$1.10 - S_3, 0)$ where S_3 is the price of the underlying in $t = 3$. We then apply a backward algorithm, which is recursive in nature, to each previous time step. We only consider the paths where we are currently in the money. For each path, we can then calculate the discounted payoff in the subsequent time steps.

Path	Y **at** $t = 2$ **(Discounted Payoff)**	X **(Stock Price at** $t = 2$**)**	**Payoff at** $t = 3$
1	-	1.08	-
2	-	1.26	-
3	0.028 $(0.07 \times e^{-0.06})$	1.07	0.07
4	0.122 $(0.18 \times e^{-0.06})$	0.97	0.18
5	-	1.56	-
6	0.310 $(0.20 \times e^{-0.06})$	0.77	0.20
7	0.244 $(0.09 \times e^{-0.06})$	0.84	0.09
8	-	1.22	-

Table 2: Discounted Payoffs and Payoffs at $t = 3$ for Each Path (Only In the Money)

We can then run a regression on these discounted payoffs using a function of our spot price to arrive at our continuation value. We can then compare this continuation value for each path to the intrinsic value of the option and record whether or not we exercise. We continue in this process until we arrive a time step $t = 0$. By the above method, we should only have one payoff for any path at one time step. Our final option valuation is simply the mean of the discounted payoffs.

1.1.2 Extension to gas storage valuation

This principle is extended in the Boogert and de Jong model [1] for gas storage valuation. The model views the gas storage as an option-like asset where the holder has the flexibility to inject, withdraw, or do nothing based on the current and expected future gas prices, operational constraints, and storage levels. Each potential decision analogous to exercising an American option at each time step, depending on which action maximizes the expected future payoff of the storage facility. Thus, the model applies Monte Carlo simulation to forecast gas prices over multiple paths and uses dynamic programming to optimize the operational strategy at each time point.

To streamline the presentation, we introduce select notation and outline the problem formulation and constraints specific to the gas storage facility. This approach helps draw parallels to the American option valuation framework discussed earlier. For a comprehensive exploration of the latter stages of the model, the reader is referred to [1].

The volume $v(t)$ at the start of day t is the initial volume and the accumulated actions up until this point.

$$v(t) := v(0) + \sum_{i=1}^{t} \Delta v(i-1) \tag{2}$$

We can then denote the payoff at date t by $h(S(t), \Delta v(t))$ where:

$$h(S(t), \Delta v(t)) := \begin{cases} -c(S(t))\Delta v(t) & \text{if inject at day } t \\ 0 & \text{if do nothing at day } t \\ -p(S(t))\Delta v(t) & \text{if withdraw at day } t \end{cases} \tag{3}$$

where $c(S(t))$ and $p(S(t))$ are the cost of injection and the profit of withdrawal respectively.

$$c(S(t)) := (1 + a_1)S(t) \tag{4}$$

$$p(S(t)) := (1 - a_2)S(t) \tag{5}$$

Note that for simplicity, we have omitted the the cost of bid-ask spreads.

We also have constraints on the volume of gas that can be stored in the facility at any time over the duration of the contract:

$$v^{\min}(t) \leq v(t) \leq v^{\max}(t) \tag{6}$$

There are further constraints on the rates at which we can withdraw and inject gas:

$$\dot{v}^{\min}(t, v(t)) \leq \Delta v(t) \leq \dot{v}^{\max}(t, v(t)) \tag{7}$$

In essence, the value of our storage contract can be seen as the expected value of accumulated future payoffs $h(S(t), \Delta v(t))$ under some optimal exercise strategy π:

$$\sup_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} e^{-\delta t} h(S(t), \Delta v(t)) + e^{-\delta(T+1)} q(S(T+1), v(T+1))\right] \tag{8}$$

where δ is our risk-free rate and $q(S(T+1), v(T+1))$ is the potential penalty on settlement.

From this point, we can draw parallels to the valuation of the American option, and proceed to look to use Least Squares regressions and Monte Carlo simulations in our gas storage valuation model.

This approach provides a robust framework for managing and valuing storage facilities under uncertain price movements, catering to both immediate operational decisions and long-term strategic planning.

1.1.3 cmdty-storage

The cmdty-storage library, written by Jake C. Fowler, provides tools for modeling and valuing commodity storage assets within the energy and commodities markets. It enables users to simulate and optimize the intrinsic and extrinsic values of storage facilities under various market and storage conditions. It does this by allowing the user to first create a *CmdtyStorage* object with the various constraints. The library supports multi-factor price modeling, as well as modelling via user defined price simulations.

The core of the library itself is written in C# with various API layers. The one we make use of is the Python API.

1.2 Price Models

We must first decide on a choice of models for the evolution of the underlying forward price and spot price.

1.2.1 One-factor model

The one-factor price model we use is governed by the following stochastic differential equation (SDE) for the spot price $S(t)$ of a commodity:

$$\frac{dS(t)}{S(t)} = k\left(\mu(t) - \ln S(t)\right) dt + \sigma\, dW(t) \tag{9}$$

where:

- $S(t)$ is the spot price of the commodity at time t.
- $\mu(t)$ is the logarithm of the mean level towards which the price reverts, possibly a function of time.
- k is the mean reversion speed, a positive constant indicating how fast the spot price reverts to its mean level.
- σ represents the volatility of the spot price, quantifying the standard deviation of the price's logarithmic returns.

- $dW(t)$ denotes the increment of a standard Wiener process (or Brownian motion), which introduces the random fluctuation into the model.

1.2.2 Three-factor model

The three-factor model introduced by Boogert and de Jong [2] and subsequently rewritten by Fowler [3] aims to better capture the seasonal variations exhibited within gas prices.

$$\frac{dF(t,T)}{F(t,T)} = \sigma_{\text{spot}} e^{-\alpha(T-t)} dz_{\text{spot}}(t) + \sigma_{\text{seas}}(T) dz_{\text{seas}}(t) + \sigma_{\text{long}} dz_{\text{long}}(t) \tag{10}$$

The forward price dynamics are influenced by three stochastic factors, each with distinct characteristics:

- **Spot Factor:**
$$\sigma_{\text{spot}} e^{-\alpha(T-t)} dz_{\text{spot}}(t)$$
This term accounts for immediate market shocks that decay exponentially with a rate α as the maturity T approaches, reflecting diminishing impacts on forward prices.

- **Seasonal Factor:**
$$\sigma_{\text{seas}}(T) dz_{\text{seas}}(t)$$
Represents seasonal (winter-summer) variability in prices with volatility $\sigma_{\text{seas}}(T)$ that potentially varies with time T, capturing the influence of predictable seasonal changes.

- **Long-Term Factor:**
$$\sigma_{\text{long}} dz_{\text{long}}(t)$$
Captures consistent long-term trends in market prices with a constant volatility σ_{long}.

We can then use the above to derive an analytical form for the spot price, $S(t) = F(t,t)$ to be used for simulation.

2 Methodology

The task is to build a simple model to predict the a given storage facility's behaviour over a 2 year period, starting April 1$^{\text{st}}$ 2024. We are also given the gas forward curve for this period.

The constraints of the storage facility are given in Table 3.

Parameter	Value
Max Withdrawal Rate	50.00
Max Injection Rate	30.00
Minimum Inventory	700.00
Maximum Inventory	4000.00
Current Inventory	750.00
Injection Cost	0.55
Withdrawal Cost	1.40

Table 3: Storage facility constraints

We construct an interest rate curve to be used for discounting using the current US yield curve [4].

We then use our one-factor and three-factor price models to simulate prices. The parameters for both models are given in Table 4 which are taken from historical Dutch TTF prices [2]. Note that whilst the cmdty-storage library has a built in function to simulate three-factor model prices and subsequent storage valuation, it does not provide the same API for a one-factor model. As a result, we use the "value_from_sims" function to build the storage valuation and write code to simulate the one-factor model from scratch.

A few things to note:

- Our storage facility has a non-zero minimum volume, and therefore we must also supply a function to deal with the valuation of the facility at the end of the contract. We set this to be 0.97 $\times$ final volume $\times$ final spot price.
- We must also convert the monthly forward curve to one which has daily granularity to be used in the model.

Model Type	**Parameter**	**Value**
One-Factor Model	Mean Reversion, κ	12%
	Spot Volatility, σ_{spot}	100%
Three-Factor Model	Mean Reversion, κ	12%
	Spot Volatility, σ_{spot}	100%
	Long-Term Volatility, σ_{long}	20%
	Seasonal Volatility, σ_{season}	20%

Table 4: Parameters for One-Factor and Three-Factor Models

3 Results

Table 5: NPV Values for Three-factor and One-factor Models

NPV Type	**One-Factor Model**	**Three-Factor Model**
Full NPV	24,895,207	19,473,308
Intrinsic NPV	44,350	44,350
Extrinsic NPV	24,850,856	19,428,958

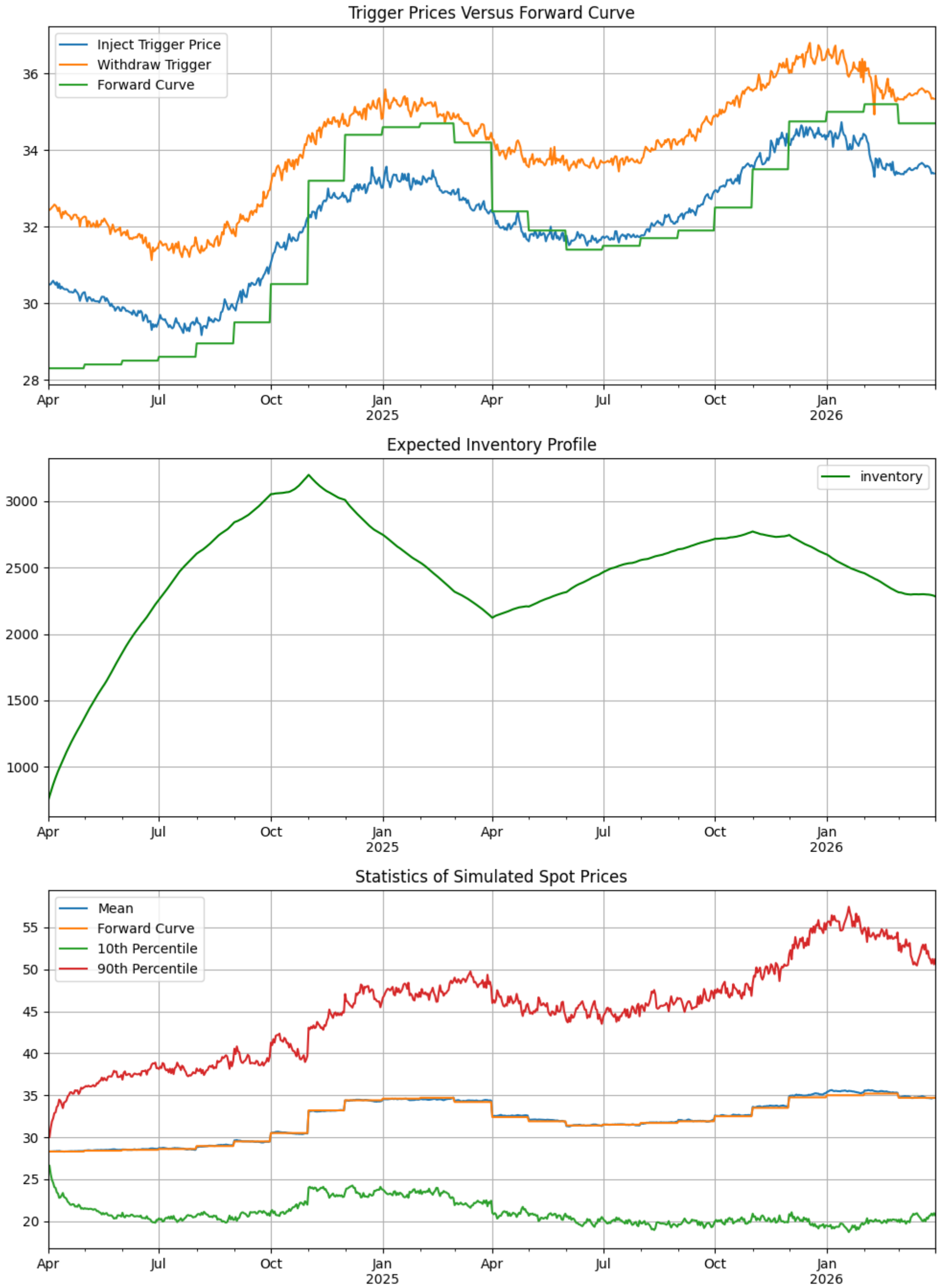

Figure 1: Results from the Three-Factor Model

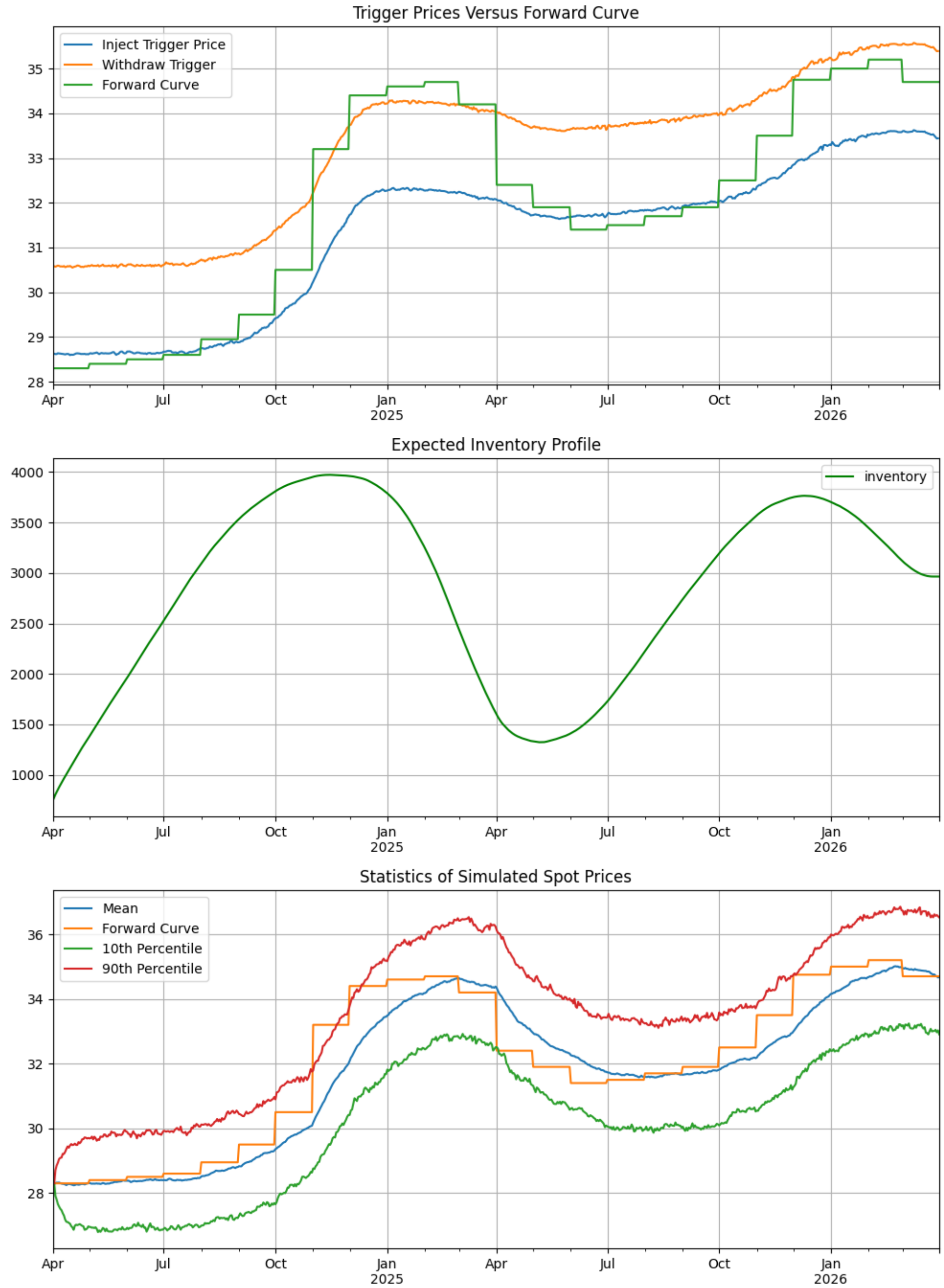

Figure 2: Results from the One-Factor Model

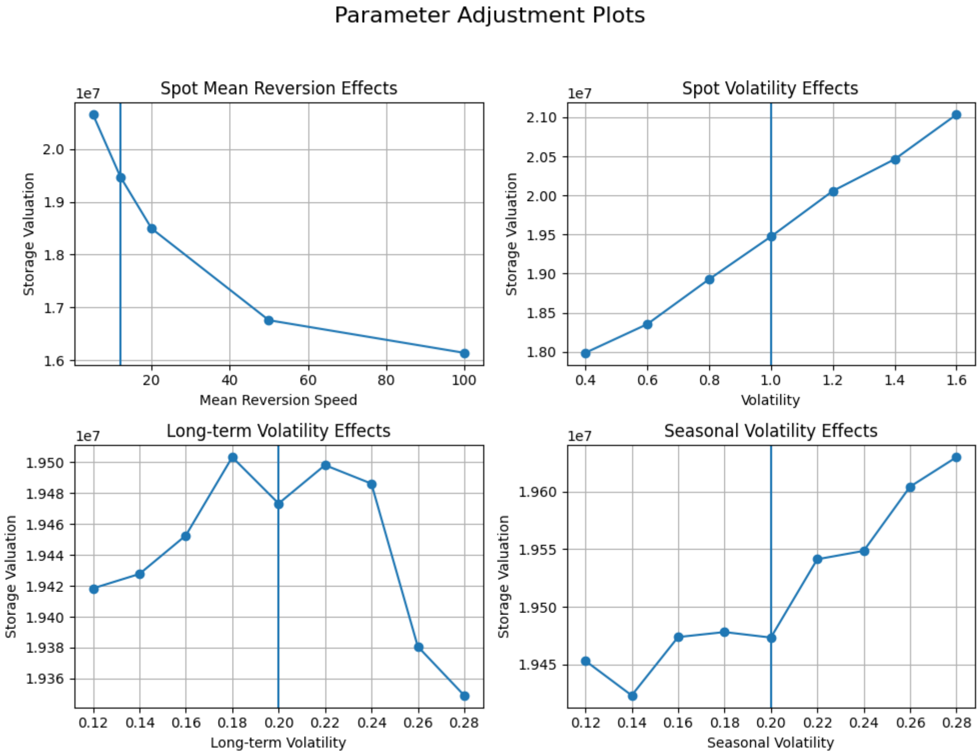

Figure 3: Parameter Adjustment Plots

4 Discussion and potential improvements

4.1 Discussion

Overall, we find that the three-factor model is better able to replicate the forward curve (which we assume to be the expected spot price). The valuation in the one-factor case is higher, but is less reliable. The final valuation in both cases, and in particular the extrinsic value is significantly higher than those shown in the examples on the library page. This can be explained by the fact that we have a non-zero minimum value. If one were to compare the behaviour of the expected inventory profile with a minimum volume of zero, we would see that in most cases after a certain point the model tries to withdraw all the way down to zero to avoid the penalty.

When we look at the change in the spot mean reversion, we find that the higher the mean reversion speed, the lower the storage valuation. This can be explained by the fact that a higher mean reversion means large swings return back to the average level faster and there are fewer opportunities to capture profit.

For both the spot and seasonal volatility, we generally see an increase in the storage value when these increase. Both can be explained similarly to the above in that higher volatility results in greater price swings which the storage facility can make profit on.

4.2 Potential improvements

The library itself is currently written mostly in C# and started off as a .NET only project. This means we have Python code calling into .NET, which is non trivial for the average Python user to then debug. An improvement in this library would be rewrite the library in Python using Numba.

One particular bug encountered was in the use of the "value_from_sims" function. The function allows for the factors in a price process to be optional arguments (as in the one-factor case, adding factor dataframes does not make sense). However, forgoing adding these parameters results in an error. A work around is to simply use the simulated price dataframes in the one-factor case as the factors as this is indeed the only factor.

4.3 Fundamental gas trading model

When looking at incorporating the gas storage valuation model into a fundamental gas trading model, one could do the following:

- Use the specific trigger prices along with the trigger prices of other storage assets the trader may hold and rank them from low to higher opportunity cost. In effect, creating an internal merit order for portfolio optimisation.
- The forward curve used in the storage valuation is assumed to be from the exchange. External fundamental data such as weather, and other potential supply-demand shocks can be incorporated to form a different view of the forward curve which could actually more closely resemble the eventual spot price in the future. Effectively we would be using our own view of the forward market instead of the one given from the exchange.

References

[1] Boogert A and C de Jong. Gas storage valuation using a monte carlo method. 2006.

[2] Boogert A and C de Jong. Gas storage valuation using a multi-factor price process. 2011.

[3] Jake C. Fowler. Three factor seasonal commodity price process. 2020.

[4] GuruFocus. U.s. treasury yield curve. https://www.gurufocus.com/yield$_{c}urve.php, Accessed :$ $[2024]. Retrieved[07-05-2024]$.

[5] Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.