Senior Design Report

Team Gas Storage:

Matthew Kindy II, Aurea Li, & Dillan Prince

April 29. 2016

Contents

1 Introduction

Through commodity trading, investors or businesses can obtain price insurance in a volatile market. For the scope of this project, the commodity that we are interested in is natural gas.

A forward contract is an agreement to buy or sell an asset at a certain future time for a certain price. When a forward contract is sold, the buyer is obligated to pay for and receive a commodity at some agreed upon time in the future. Likewise, the seller of the forward contract is under obligation to sell and thus deliver the commodity. Future contracts are forward contracts that are traded on the exchange, which provides a mechanism that gives the two parties a guarantee that the contract will be honored. Futures are used as a hedging tool; as the price of natural gas is constantly changing, future contracts are bought and sold in order to lock down costs and revenues for natural gas.

The process of trading natural gas can be simplified down to the following: a forward contract is bought (and thus the buyer receives natural gas at a specified price). The natural gas is then delivered and stored in a storage facility at the time the forward contract is exercised. Then, a forward contract is sold and the company delivers the natural gas to the new buyer. By choosing appropriate prices for buying and selling forward contracts in natural gas, profits can be made off of receiving and delivering this commodity. However, future prices change on a continuous basis and behave somewhat unpredictably. Moreover, companies have to factor in the cost of transporting the gas as well as storing it.

We are interested in building a mathematical optimization model that can determine the forward-looking valuation of natural gas storage assets and the trading strategies used to lock in the values.

2 Literature Review

Within existing literature on natural gas storage optimization, most models seem to use one of two main techniques: dynamic programming or specially designed heuristic techniques [1].

Approaches to natural gas storage optimization are highly dependent on how the futures market is perceived and formulated mathematically. One implementation, called the rolling intrinsic approach [2] involves assuming that the holder captures the intrinsic value at the start of the forward contract, and decides whether to change or fix his/her position by comparing the profit of doing so to the transaction costs. Repeated maximization is then performed on this intrinsic value using principal component analysis to determine loads from forward curve movements, and then Monte Carlo simulations to model possible realizations [3]. Unfortunately, the rolling intrinsic approach is not guaranteed to be profit maximizing [2]. A similar method expands on the rolling intrinsic approach, accounting also for spot price, which is then implemented with Monte Carlo ordinary least square method [4]. This has the advantage of allowing for ease of incorporation of additional constraints. However, the method is relatively slow, so it requires greater attention to computational decisions [2].

A different class of methodologies involve only current spot price in their evaluations rather than the entire forward contract price curve. Stochastic dual dynamic programming is one such approach [5]. One particular model in this class of methodologies utilizes a discrete-time stochastic dynamic programming for the evolution of the forward curve model [6]. This results in nearly optimal, although not guaranteed optimal, upper and lower bounds on the valuation of contracts. Nonetheless, the strategy is fairly unpopular among natural gas storage traders due to the belief that the price models used either are not or cannot be easily calibrated to be consistent with the hedging strategy that accompanies the contract strategy [6]. Unfortunately, the major alternative, using high-dimensional forward models (as is natural to do within linear programs), is impossible to combine with this stochastic method, which makes the approach undesirable for our model.

Another approach involves comparing gas storage to options, and a model is derived from methodologies stemming from option pricing. By ignoring operating characteristics, such as the available injection and withdrawal rates (injection capacity and deliverability), and focusing on the natural gas as an underlying asset, a natural gas well is seen as a series of call options on the price of natural gas where the strike price is calculated as the total production costs. However, difficulties can arise with this approach. In particular, reintegrating operating characteristics into the model can be challenging, and extreme price fluctuations can be difficult to model around. [1] Furthermore, while there are a few approaches for pricing general options, which include using numerical PDE solvers to solve the optimization model, using Monte-Carlo simulations, and tracing binomial-trinomial trees [1, 7], Monte-Carlo simulations do not provide the means to develop an optimal purchase and sale strategy, and binomial-trinomial trees have various issues which specifically inhibit them from being used for natural gas options pricing [1]. Thus, there remains only one option which is tenable in this regard – numerical PDE solvers.

For the purposes of our model development, this restriction to numerical PDE solvers can be disregarded until we begin to derive options-based natural gas strategies since the initial focus is on forward-contract based trading strategies. At early stages, a algorithmically greedy daily rolling method is utilized for our multiday trading scenario because it is easily extensible in terms of complexity and requires the least natural gas domain expertise, taking advantage of the entire forward curve.

3 Linear Simplified Model

Our approach focused on an iterative design, building a model starting from a simplified base. The main reasoning for doing this is to leverage the relative simplicity of verification and validation of simple models to ensure our model rests on a solid mathematical base. After each model iteration is verified and validated, an additional layer of complexity will be added. The first iteration of the model involves simplification of the problem in order to model it with a linear program (LP).

The goal of this LP-based model is to maximize profit obtained through buying and selling forward contracts for natural gas, subject to constraints such as minimum and maximum allowed inventory level of storage, required starting and ending inventory levels, and rates of injection and withdrawal of natural gas.

3.1 Variable definitions

In our model, we will be using the following variables and constants:

$g \in \mathbb{R}$: A constant conversion factor from contracts to MMBTU. Natural gas futures contracts are commonly standardized for a total delivery of 10,000 MMBTU per contract over a given delivery month, where each delivery month has a varying number of days.

One MMBTU is equal to one million British Thermal Units (BTU), which is a measurement of energy equivalent to 1.06 million Joules, and is the industry standard.

$N \in \mathbb{R}$: The number of months over which we will be optimizing.

$d \in \mathbb{R}_+^N$: Vector of forward contracts sold, where d_k is the number of forward contracts sold during month k.

$e \in \mathbb{R}_+^N$: Vector of forward contracts bought, where e_k is the number of forward contracts exercised during month k.

$f \in \mathbb{R}^N$: Vector of forward curves, where f_k is the forward curve for month k.

A forward curve is a list of prices for forward contracts at some moment in time in the future.

$q \in \mathbb{R}^N$: Vector of withdrawal costs, where q_k is the cost of withdrawal for month k. ($)

$p \in \mathbb{R}^N$: Vector of injection costs, where p_k is the cost of injection for month k. ($)

$i \in \mathbb{R}^N$: Vector of maximum injection rates, where i_k is the maximum injection rate for month k. (MMBTU per day)

$w \in \mathbb{R}^N$: Vector of maximum withdrawal rates, where w_k is the maximum withdrawal rate for month k. (MMBTU per day)

$\ell \in \mathbb{R}^N$: Vector of minimum inventory requirements, where ℓ_k is the minimum inventory required at the end of month k. (MMBTU or % of total capacity)

$u \in \mathbb{R}^N$: Vector of maximum inventory allowance, where u_k is the maximum inventory allowed at the end of month k. (MMBTU or % of total capacity)

$\theta \in \mathbb{R}^N$: Vector of days per month, where θ_k is the number of days in month k.

While this may seem like an obvious value, since a month always has between 28 and 31 days, it is not necessarily true that the first month of operation is January, and the number of months over which we are optimizing is variable. Use of this variable allows us to generalize to any starting month and any length of time.

$\phi \in \mathbb{R}^N$: Vector of conversion values for MMBTU per day for each month, where ϕ_k is the quantity of MMBTU per day per contract for month k. Specifically, ϕ_k is defined to be g/θ_k.

$v \in \mathbb{R}^n$: Vector of inventory level, in MMBTUs, in storage. Here, n $= \sum_i^N \theta_i$.

In order to convert from months-and-days notation (where an arbitrary day of the time period is represented as the j-th day of the k-th month) to solely days notation (where an arbitrary day of the time period is represented simply as the s-th day of the time period), we will be using the following function $s : \mathbb{R}^2 \mapsto \mathbb{R}$:

$$s(j,k) = j + \sum_{i=1}^{k-1} \theta_i$$

This allows us to convert from $v_{j,k}$, the inventory level on the j-th day of the k-th month, to $v_{s(j,k)}$, the inventory level on the s-th day.

3.2 Objective function

In order to maximize profit, we must maximize the difference between revenue and cost. Revenue is earned when contracts are sold, and the amount earned in revenue in a given month k is equal to the product of the quantity of forward contracts sold (d_k) and the price of forward contracts (f_k). Hence, revenue (R) is calculated as follows:

$$
\begin{aligned}
R &= \sum_{k=1}^{N} d_k f_k \\
&= d^T f.
\end{aligned}
$$

We incur costs in several ways: injecting gas from storage, withdrawing gas from storage, and purchasing contracts. Similar to selling contracts, the cost associated with purchasing a contract during month k is equal to the product of the quantity of forward contracts bought (e_k) and the price of forward contracts (f_k). When calculating the cost of injecting or withdrawing natural gas for a month, we convert from contracts to MMBTU using the constant conversion factor so the cost of injection and withdrawal for month k is the product of the amount of contracts bought or sold (e_k or d_k) with the injection or withdrawal costs (p_k or q_k) and the constant conversion factor (g). The cost (C) is then represented as the sum of the total spent on purchasing contracts, $\sum e_k f_k$, the total cost of injecting gas, $\sum g e_k p_k$, and the total cost of withdrawing gas $\sum g d_k q_k$. More specifically, cost is represented as

$$
\begin{aligned}
C &= \sum_{k=1}^{N} g d_k q_k + \sum_{k=1}^{N} e_k f_k + \sum_{k=1}^{N} g e_k p_k \\
&= g d^T q + e^T f + g e^T p.
\end{aligned}
$$

The optimal profit is the maximum possible difference between revenue, R, and cost, C. To form our model in standard form, we find

$$
\begin{aligned}
P &= \max R - C \\
&= \max_{d,e\in\mathbb{R}_+} \left(d^T f - (g d^T q + e^T f + g e^T p)\right) \\
&= \max_{d,e\in\mathbb{R}_+^N} (d^T f - g d^T q - e^T f - g e^T p) \\
&= \max_{d,e\in\mathbb{R}_+^N} \left[d^T (f - gq) + e^T(-f - gp)\right] \\
&= \max_{d,e\in\mathbb{R}_+^N} \begin{pmatrix} f - gq \\ -f - gp \end{pmatrix}^T \begin{pmatrix} d \\ e \end{pmatrix} \\
&= \max_{x\in\mathbb{R}_+^{2N}} c^T x.
\end{aligned}
$$

Where

$$c = \begin{pmatrix} f - gq \\ -f - gp \end{pmatrix} \in \mathbb{R}^{2N}$$

and

$$x = \begin{pmatrix} d \\ e \end{pmatrix} \in \mathbb{R}_+^{2N}.$$

3.3 Constraints

Let j be the day index and k be the month index for the j-th day of the k-th month of the time period. Let the conversion function $s : \mathbb{R}^2 \mapsto \mathbb{R}$ be the s-th day of the time period, where

$$s(j,k) = j + \sum_{i=1}^{k-1} \theta_i.$$

Then the set of physical constraints on the optimization problem are as follows:

- Inventory minimum: There is a minimum amount of natural gas that must be in storage at the end of the last day of each month, or, equivalently, at the beginning of the first day of each month. Therefore, the inventory minimum for the k-th month must be less than the inventory level at the beginning of the first day of the $(k+1)$-th month. This can be expressed as

$$\forall k \in \{1, 2, ..., N\}, \quad \ell_k \leq v_{s(1,k+1)}.$$

- Maximum injection: There is a limit to how much natural gas can be injected into natural gas storage each day. For all months, the difference between the inventory level, v, for two consecutive days of the k-th month must be less than the maximum injection rate for that month, i_k. This can be expressed as

$$\forall k \in \{1, 2, ..., N\} \text{ and } \forall j \in \{1, 2, ..., \theta_k - 1\}, \quad v_{s(j+1,k)} - v_{s(j,k)} \leq i_k.$$

- Maximum withdrawal: There is a limit to how much natural gas can be withdrawn from natural gas storage each day, dependent on the month. For all months, the difference between the inventory level, v, for two consecutive days of the k-th month must be greater than the negation of the maximum withdrawal rate for that month, w_k. This can be expressed as

$$\forall k \in \{1, 2, ..., N\} \text{ and } \forall j \in \{1, 2, ..., \theta_k - 1\}, \quad v_{s(j+1,k)} - v_{s(j,k)} \geq -w_k,$$

or alternatively,

$$\forall k \in \{1, 2, ..., N\} \text{ and } \forall j \in \{1, 2, ..., \theta_k - 1\}, \quad v_{s(j,k)} - v_{s(j+1,k)} \leq w_k.$$

- Starting and ending inventory levels: There is an inventory contained within storage at the beginning of the first day, and there must be a specified inventory level in storage at the end of the last day. Take V_0 to be the inventory level on the first day of the time period, and take V_N to be the inventory level that must be in storage at the end of the last day of the time period. Then $v_{s(1,1)}$ is equal to V_0 and $v_{s(\theta_N,N)}$ is equal to V_N.

$$v_{s(1,1)} = V_0.$$

$$v_{s(\theta_N,N)} = V_N.$$

4 Modeling Methods

4.1 MATLAB Linear Programming on Simplified Problem

The original problem involved daily withdrawal and injection constraints which were based on functions of the inventory level. Since the inventory level is guaranteed to be a function of future contracts bought and sold, this creates a non-linear problem. As a result, we reduced the complexity by assuming constant daily withdrawal and injection constraints for each month. This leads us to a simplified linear version of the optimization problem.

It should be noted that this simplification does not lead to an artificial problem. Due to the desire for operational simplicity, storage assets are often modelled in this way in industry.

4.2 Constraint Generation

In the reduced problem, two matrices need to be constructed to properly solve this problem. The first is A, which is constructed such that $Ax \leq b$. The second is A_{eq}, which is constructed such that $A_{eq}x = b_{eq}$. b and b_{eq} are both vectors.

The constraints in $Ax \leq b$ are row-by-row generated in MATLAB. The first set of constraints in A is derived from

$$\forall k \in \{1, 2, ..., N\} \quad \ell_k \leq v_{s(1,k+1)}$$

which is equivalent to

$$\forall k \in \{1, 2, ..., N\} \quad -v_{s(1,k+1)} \leq -\ell_k.$$

where, in terms of contracts for each month,

$$v_{s(1,k+1)} = \phi_{k+1}(e_{k+1} - d_{k+1}) + g \sum_{n=1}^{k} (e_n - d_n).$$

$$\forall k \in \{1, 2, ..., N\} \quad v_{s(1,k+1)} \leq u_k.$$

In other words, the inventory level at a given day is the sum of the net number of contracts of the prior complete months multipled by the constant conversion rate from contracts to MMBTU, g, plus the remaining days multiplied by the conversion rate from contracts to MMBTU/day for month k, ϕ_k. This set of constraints is a limit on the minimum gas holdings in storage at the end of the last day of each month. The coefficients for each month's set of contracts can then easily be extracted and used as the entries for a row of A.

The second set of constraints in A is derived from the maximum injection and maximum withdrawal constraints. Since this simplified linear program involves *constant* monthly injection and withdrawal limits, the constraints can be rewritten by month:

$$\forall k \in \{1, 2, ..., N\} : \begin{cases} \phi_k(e_k - d_k) \leq i_k, \\ \phi_k(d_k - e_k) \leq w_k. \end{cases}$$

The first limits the daily injection rate for each month to i_k (in MMBTU), where the injection rate is easily expressed here as the net number of contracts exercised in a month multiplied by the constant conversion from contracts to MMBTU/day for month k, ϕ_k.

The second limits the daily withdrawal rate for each month to w_k (in MMBTU), where the withdrawal rate is easily expressed here as the net number of contracts sold in a month multiplied by the constant conversion from contracts to MMBTU for month k, ϕ_k.

The constraints in $A_{eq}x = b_{eq}$ are set from the boundary conditions, where the initial inventory level v_1 is equal to V_0 and the final inventory level v_n is equal to V_N, where n is the sum of all the entries in θ.

However, it is simpler to describe these constraints by a single range constraint by using contract-days:

$$\sum_{k=1}^{N}(e_k - d_k) = \frac{V_N - V_0}{g}$$

$$\sum_{k=1}^{i}(e_k - d_k) = \frac{V_i - V_0}{g}$$

Here, it is asserted that the sum of the net contracts is equal to the difference in boundary inventory levels divided by the constant conversion from contracts to MMBTU, resulting in contract equality. Thus, the two equality constraints are accurately represented as a single equality constraint with clear coefficients for A_{eq}.

4.3 Solution Validation

Some basic validation was performed on the output of the MATLAB function that was written. In particular, it was verified against a set of carefully selected inputs (within their respective domains) that the output of the optimization process were non-negative vectors d and e and a non-negative scalar f (the profit calculated from the selection of d and e).

It was found that the solutions provided through the algorithm satisfied validity requirements insofar as all vector elements were nonnegative for all of 10000 carefully selected checks. The variables used in construction of the LP were sampled from the distributions as described below:

5 Daily Rolling Trading

The next iteration of our model sought to allow for determination of an optimal strategy for trading in the face of new information. In particular, although the original model optimizes over a given forward curve, the values of this forward curve can be volatile on a day-to-day basis. Therefore, it may be possible to increase the value of a given forward contract position by re-evaluating for these changes in price.

Our model approaches this problem by implementing a smart rolling intrinsic algorithm, which follows a greedy algorithmic heuristic. The algorithm can be outlined in the following way:

1. Optimize a trading position for a given forward curve.

2. Once new market information is provided, calculate an optimal trading position for that new forward curve.

3. Adjust the initial trading position to match the newly calculated optimal position, dependent on whether costs of doing so are outweighed by profit of doing so.

This algorithm was successfully implemented and included in the next iteration of our model, allowing the model to return a series of decisions for a given set of forward curves.

Here is a comparison of the objective values over time for the Linear Simplified Model (LSM) and the Daily Rolling Model (DRM). Note that for each iteration of the forward curve, the objective values calculated for LSM are independent of each other. Initially both start off with the same objective value. For iteration number 2, however, the objective value calculated for the LSM jumps up to roughly 525 as opposed to the previous value of 460 based on the forward curve data from the previous iteration. Given the new forward curve, since a higher objective value can be reached, the objective value for the DRM is adjusted and becomes 465. Independent optimization of the next two forward curve iterations result in lower objective values. These lower objective values calculated for LSM don't necessarily imply, however, that it is not possible to adjust and improve the objective values for DRM. This is reflected in the last iteration, where despite the new calculated objective value of LSM yielding 480, the objective value of DRM increases from 465 to 471.

It should be noted that the DRM is independent of the optimization model chosen. In other words, it is a technique that can be applied with more or less sophisticated optimization models, dependent on what is desirable.

6 Piecewise Linear Constraints

6.1 Convex Hull Relaxation

Our next step is to implement additional constraints that may not necessarily be linear. For example, earlier we assumed that the injection and withdrawal rates were constant within each month. In other words, injection and withdrawal rates only varied by month. Realistically, however, real-world injection and withdrawal rates do take the current inventory level into account and these rates are typically described by piecewise linear functions. Such functions may not necessarily be convex in nature. One solution to this issue is to take the convex hull of these constraints. We are thus optimizing over a relaxation of the original problem. This relaxation will yield us solutions that are guaranteed to be feasible for the set of constraints from which the convex hull was derived from.

6.2 Branch and Cut

As we further refine our model, we are interested in implementing more complex constraints in which our "convex hull" solution may not necessarily be globally optimal or even feasible. The algorithm *branch and cut* (B&C) is used to approach these constraints. B&C is most commonly used to solve integer linear programs. Let P be the set of all our linear constraints and Ω the set of all nonlinear constraints. Given the following piecewise linear constraint problem:

$$\pi = \max \{c^T x : x \in P, x \in \Omega, x \geq 0\}$$

we first consider the relaxation linear program

$$\Pi = \max \{c^T x : x \in P, x \geq 0\}.$$

When the B&C approach is applied to our model, Ω is our set of monthly injection and withdrawal constraints. These are not necessarily nonlinear, however, as workers generally obey piecewise linear constraints for injection and withdrawal in reality, rather than one of the numerous real gas laws. The issue that arises is that, after integration to force continuity of our constraints, these piecewise linear constraints remain nonconvex, and thus require us to relax our constraints to form a convex hull. This is illustrated in Figure 4.

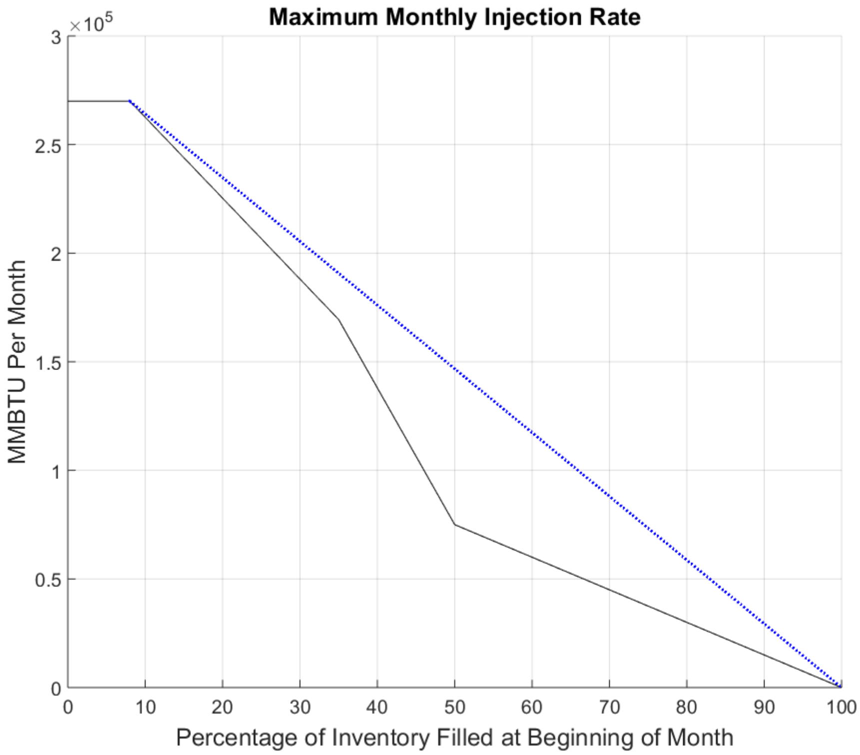

Figure 1: Taking the convex hull of our original constraints.

We solve Π, yielding an optimal value and solution of $z(\tilde{x})$ and $\tilde{x}$, respectively. If $\tilde{x}$ is a feasible solution for π, then we are done. Otherwise, we branch π by selecting an i such that $\tilde{x_i}$ violates our constraint.

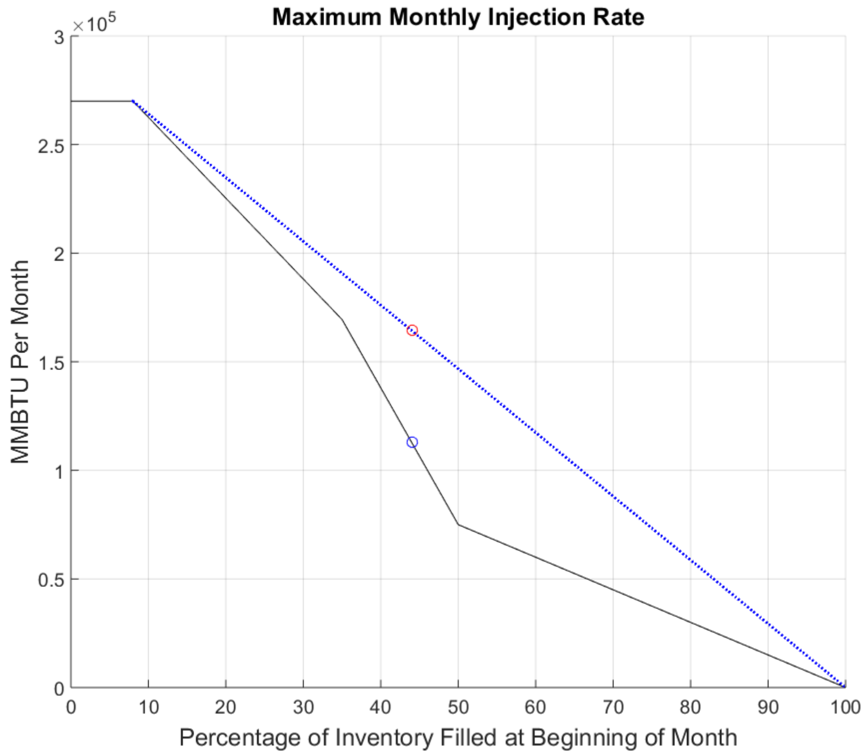

Figure 2: Checking feasibility of our relaxed LP.

In this example, $z(\tilde{x})$ and $\tilde{x}$ (shown as the red dot) are not feasible solutions.

Then, we formulate the following LPs by adding cutting planes:

$$\pi_\ell : \max\{c^T x : x_i > \tilde{x}_i, x \in P, x \geq 0\}$$

$$\pi_r : \max\{c^T x : x_i < \tilde{x}_i, x \in P, x \geq 0\}.$$

In this particular example we split the LPs based on the x coordinate of the optimal solution.

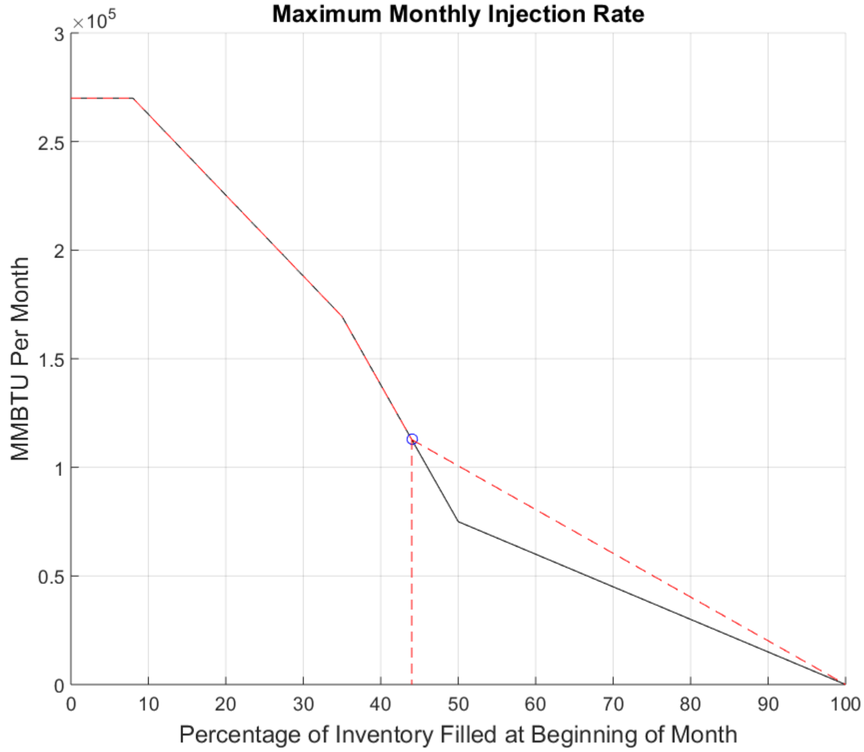

Figure 3: Splitting the LP into two problems and repeating the process.

Looking back at Figure 3, if we let I be the injection rate and V be the inventory level, the maximum monthly injection rate can be plotted using V vs I. Since the change in inventory levels from month to month is simply the injection rate, the constraints for injection and withdrawal can be written as functions of the inventory level:

$$-W(V) \leq \delta V \leq I(V) \tag{1}$$

where the change in inventory for the month $i, \delta V_i$ is $\Delta_i X$, where Δ_i is the vector of length $2N$ with value g at the ith entry and $-g$ at the $(i + N)$ entry, with zeros elsewhere. The functions W and I are piecewise linear where the jth piecewise interval takes the form

$$I_j(V) = m_j V + b_j^I$$

.

Here, m is $\frac{\Delta I}{\Delta V}$, calculated from the endpoints of the interval. It's also important to note that the inventory level for month i can be formulated as

$$V_i = s_i X + V_0$$

.

where X is the decision vector $[d \quad e]$, and s_i is the vector representing the summation of the first $i-1$ months of injection and withdrawal, or

$$\sum_{j=1}^{i-1} \Delta_j$$

.

For example, s_3 for a 4 month period would be

$$\begin{pmatrix} g & g & 0 & 0 & -g & -g & 0 & 0 \end{pmatrix}.$$

Now the constraints for a particular inventory level as in (1) can be rewritten as

$$-(m_j^W (s_i X + V_0) + b_j^W) \leq \Delta_i X \leq m_j^I (s_i X + V_0) + b_j^I, \tag{2}$$

and these can be rewritten in standard form as

$$(\Delta_i - m_j^I s_i) X \leq m_j^I V_0 + b_j^I \tag{3}$$

and

$$(\Delta_i - m_j^W s_i) X \leq m_j^W V_0 + b_j^W. \tag{4}$$

After splitting the LP into two problems, we choose one LP to reoptimize over. In Figure 7, we choose to reoptimize over the right side. The process is then repeated.

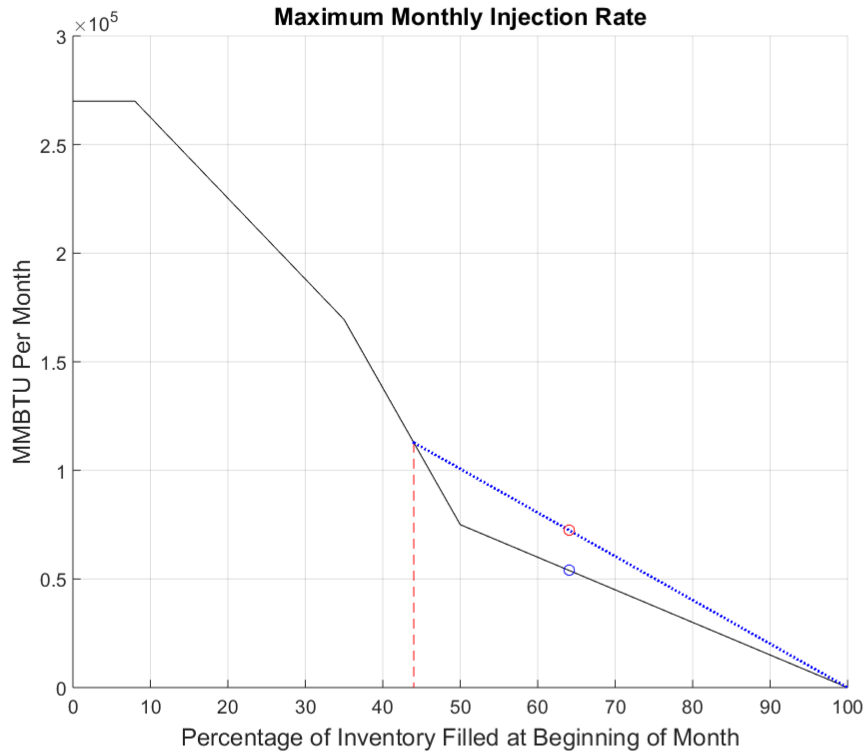

Figure 4: Reoptimizing and checking feasibility of one of the new LPs.

Note that if we take π to be a vertex and π_ℓ and π_r to be children of π, we can visualize the B&B process as a tree where each vertex represents an LP. We repeat this process with each of the LPs until one of the following occurs, in which case we prune the LP by not branching on it.

1. LP is infeasible.
2. LP optimal value is below the lower bound.
3. LP yields an optimal solution such that $x \in \Omega$. In which case we check to see if it is the most optimal solution found for π so far. If so, our lower bound is updated with the corresponding value.

Once we have successfully explored and solved all nodes (LPs), we return the lower bound as the most optimal solution.

After finding an optimum value, $z(\tilde{x})$, it is possible that $\tilde{x}$ violates one of the original non-convex injection or withdrawal constraints. In the case that such a constraint is indeed violated, we branch on that constraint and reoptimize. We repeat the B&C process as mentioned above, and are able to guarantee that the optimum value, $z(\tilde{x})$, lies within our nonconvex feasible region.

7 Model Validation

7.1 Basic Verification

A sample problem with selected inputs and known optimal solution was used to test both the LSM and the PLM:

"Optimize over the 12 month forward curve starting March 1st with contract values of \$3.25, \$3.50, \$3.75, \$4.00, \$4.25, \$4.50, \$4.75, \$5.00, \$5.25, \$5.50, \$5.75, and \$6.00 per MMBTU. There is a \$0.01 cost per MMBTU for injection or withdrawal. Daily injection limits for each month are set to 9,000 MMBTU. Daily withdrawal limits for each month are set to 6,400 MMBTU. There are 200,000 MMBTU in storage to begin, and there must be 100,000 MMBTU in storage to end. The minimum inventory level is 0 MMBTU. The maximum inventory level is the same as the capacity of 1,000,000 MMBTU."

First, this was applied to the LSM to ensure that the LSM was achieving the correct solution. The solution that was generated is displayed in Figure 5.

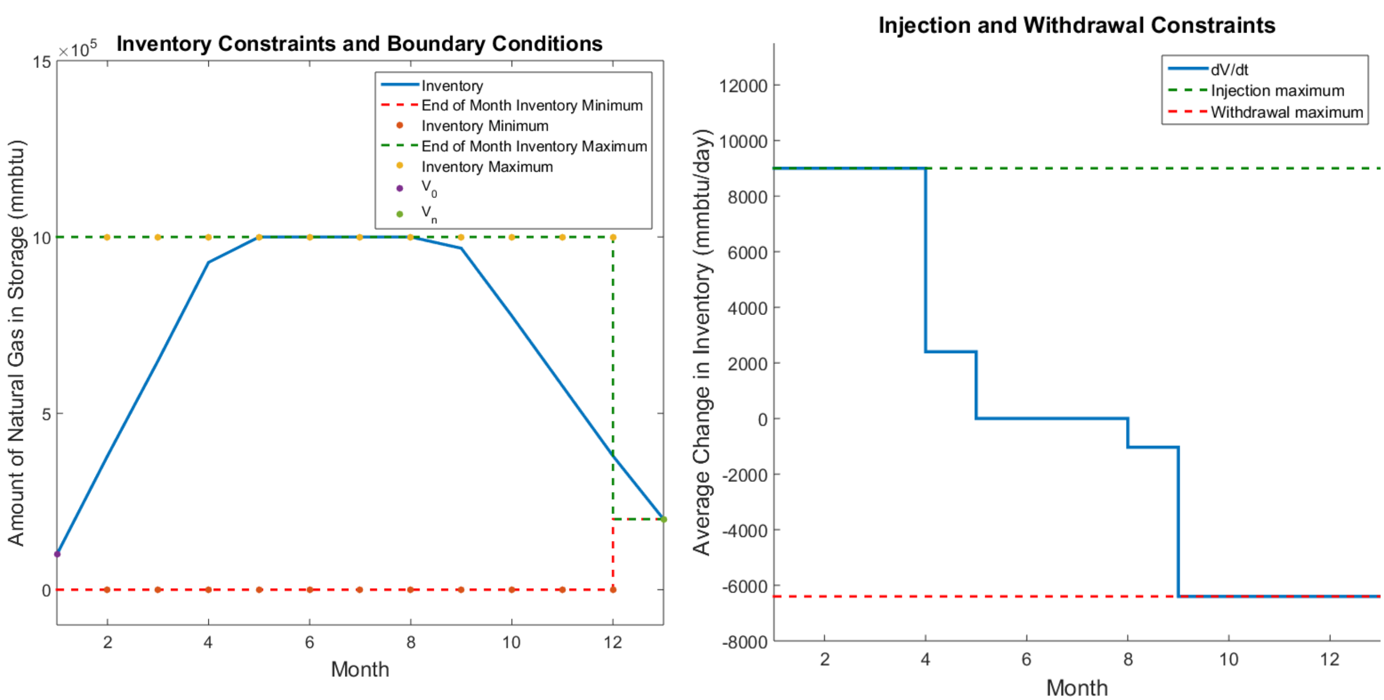

Figure 5: Visualization of minimum inventory constraints and equality constraints with LSM solution (left), and visualization of maximal injection and withdrawal constraints with time derivative of LSM solution (right) for a simple storage problem.

The numerical solution yields the decision vector components

$$d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 13.20 & 19.20 & 19.84 & 19.84 & 17.92 \end{bmatrix}$$

$$e = \begin{bmatrix} 27.90 & 27.00 & 25.10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which correspond to net contracts sold of

$$net = \begin{bmatrix} -27.90 & -27.00 & -25.10 & 0 & 0 & 0 & 0 & 13.20 & 19.20 & 19.84 & 19.84 & 17.92 \end{bmatrix}$$

First, it should be noted that this agrees with the trading intuition to "buy low and sell high." The majority of contracts are purchased when the prices are lowest (i.e. closest to the beginning of the time period). Correspondingly, contracts are sold when the prices are highest (i.e. nearest to the end). Therefore, because there is no cost other than purchasing contracts in this problem, the total profit made is simply the inner product of the net contract vector and the monthly price vector (which is simply the price per MMBTU scaled by the number of days per month). Thus, profit is calculated to be $2,165,200.

7.2 Optimization Validation

For larger and more complex constraints and solutions, it becomes more difficult to calculate that the model is appropriately designed to adhere to the imagined constraints in a way that provides an optimal solution. In order to better evaluate the model, monthly contract data was transformed into daily inventory level data to visualize the constraints and to graphically check for adherence to the constraints based on our understanding of the relationship between daily inventory level and monthly contracts.

In each of the visualizations below, it can be verified that the minimum inventory constraints and maximum injection and withdrawal constraints were satisfied by the solution.

7.2.1 General Qualitative Constraint Evaluation

Qualitative checks for validity of the LP construction and solution are very helpful in error-checking for our model. For random constructions of problem scenarios above, plots were made to assist in this effort. In addition, the use of a "realistic" problem with more complicated logistical constraints (see Figure 6) was employed to ensure that constraints were being modelled appropriately.

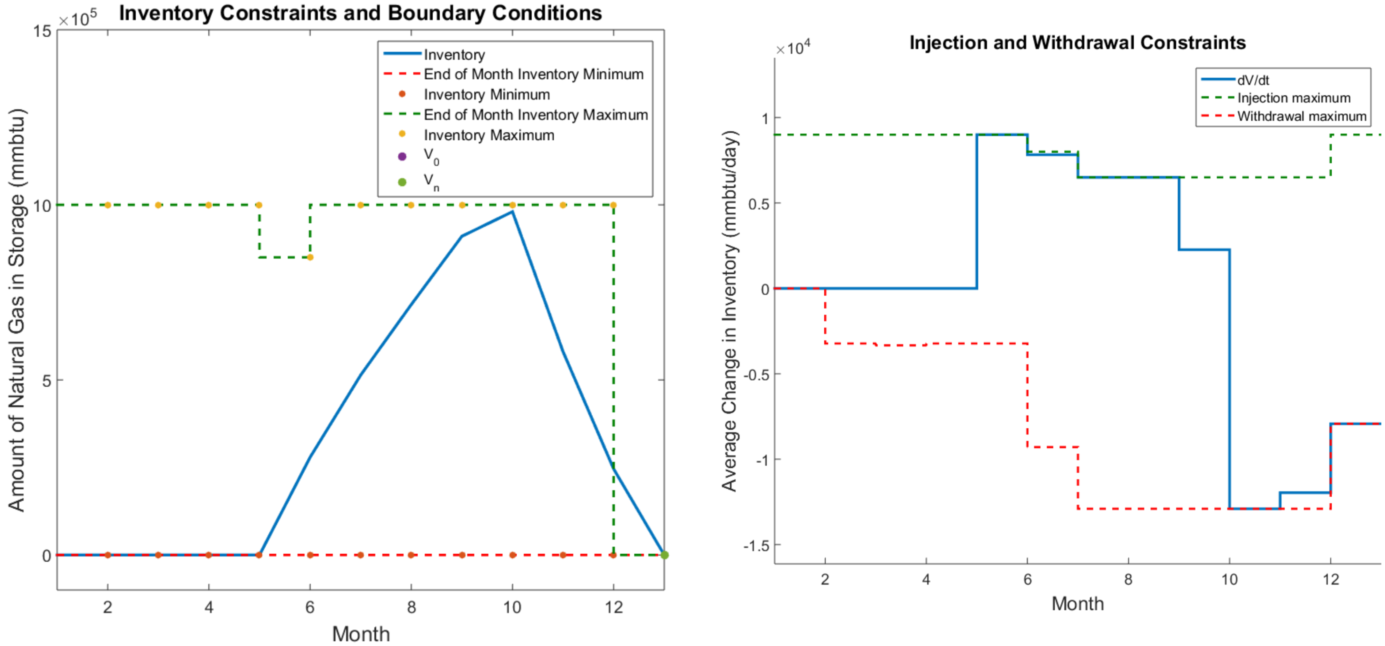

Figure 6: Visualization of minimum inventory constraints and equality constraints with solution (left), and visualization of maximal injection and withdrawal constraints with time derivative of solution (right).

Here we show that the calculated optimal solution obeys both the minimum and maximum inventory constraints, as well as the maximum injection and withdrawal constraints.

7.2.2 Minimum and Maximum Inventory Constraints

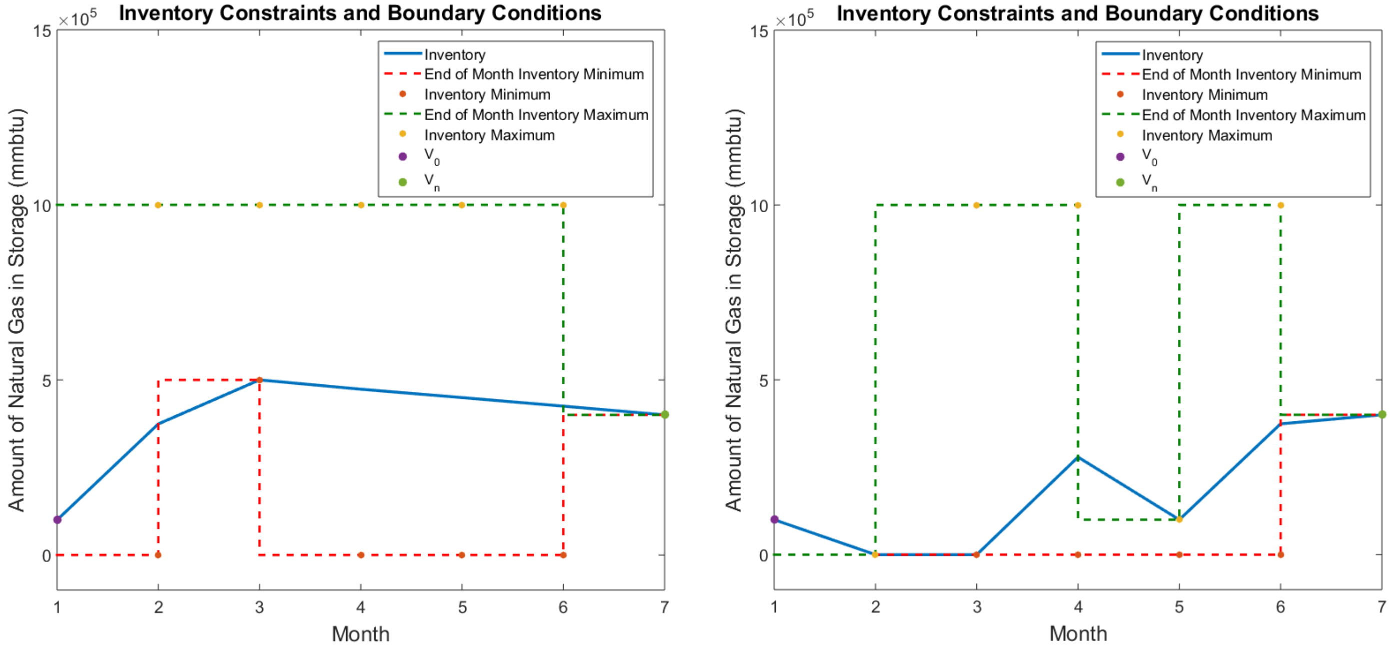

Figure 7: Minimum inventory constraint inventory level visualization (left), and maximum inventory constraint inventory level visualization (right) for verification of constraint enforcement.

The minimum and maximum inventory levels are given as constraints for each month and represented as dotted red and green lines, respectively. The blue line represents the current inventory level. Note that for both graphs in Figure 7, we see that at no month does the inventory level exceed the maximum inventory requirement for any given month. This is also true for minimum inventory requirements. Hence, the inventory level satisfies the minimum and maximum inventory constraints throughout the months, following our expectations of what we anticipate our model to show.

7.2.3 Equality Constraints

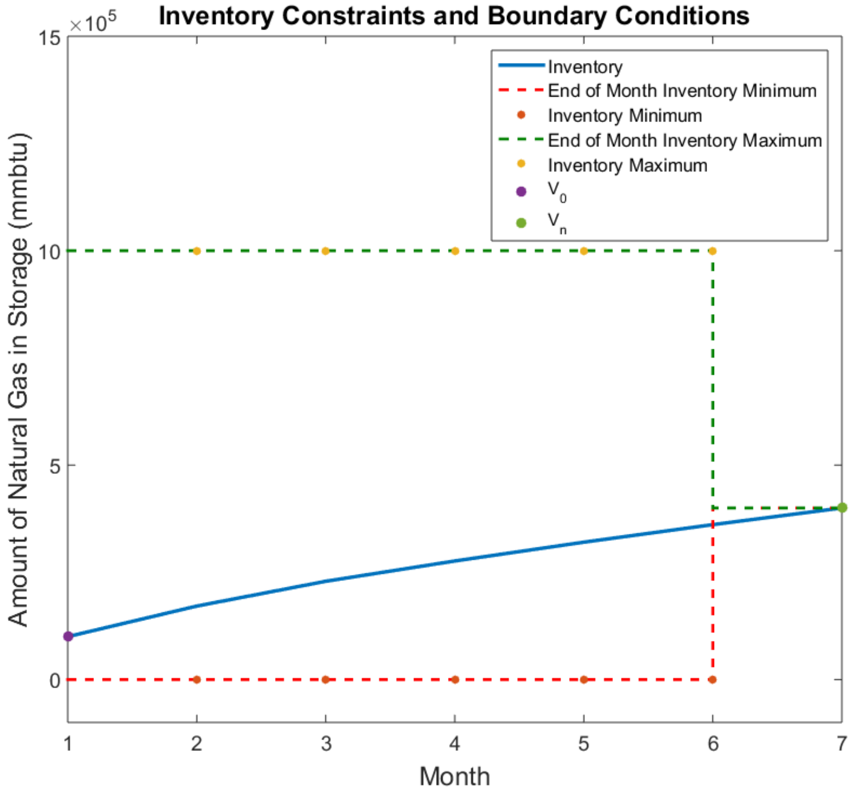

Figure 8: Inventory level visualization for verification of equality constraint enforcement.

The equality constraints the model must satisfy in this example is to have the beginning and end inventory to be at specific levels. These levels are represented as V_0 and V_n, respectively. Hence for our validation graphs, we expect that at the first month, the inventory level is equal to V_0, and at the last month, the amount of natural gas is equal to V_n. As we expect, Figure 8 follows these rules and thus satisfies the equality constraints.

7.2.4 Cross-Model Validation

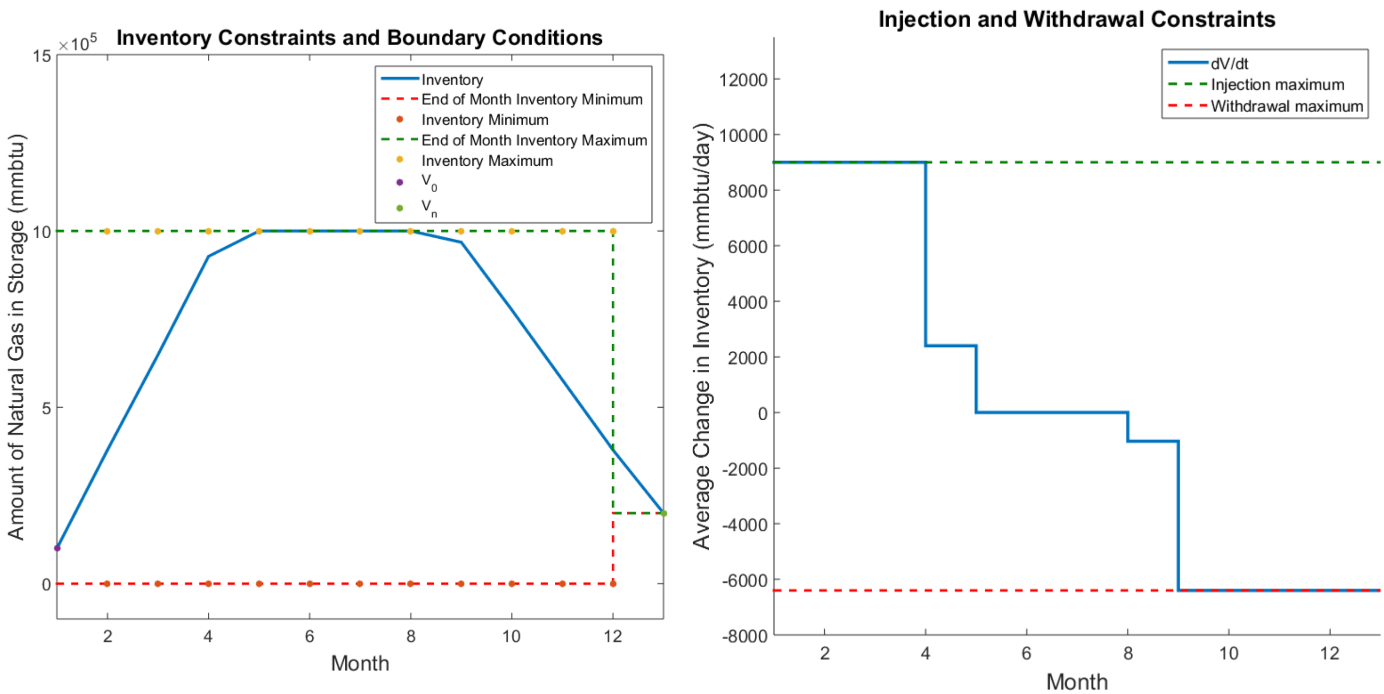

Figure 9: Visualization of minimum inventory constraints and equality constraints with PLM solution (left), and visualization of maximal injection and withdrawal constraints with time derivative of PLM solution (right) for a simple storage problem.

Another form of validation was to test the same simple problem on both the LSM and PLM model. We should expect the models to yield the same values. The above graph visualizes the inventory level over time in relation to the constraints using the PLM model. Note that Figure 9 is identical to 5. It follows given the same simple problem, both models yielded identical inventory levels for each month.

8 Conclusion

The overall problem we are trying to solve is to determine the forward-looking valuation of natural gas storage assets and the trading strategies used to lock in values by trading forward contracts. We have developed and implemented a mathematical linear program model to optimize buying and selling of forward contracts for this problem, based on the assumption that the injection and withdrawal rates are constant within each month, and later on, are only dependent on inventory level.

In the first simplified model, the maximum daily injection and withdrawal constraints are constant for each month. In reality, these specific constraints change continuously as a function of the current inventory level, since more power is required to inject natural gas into storage as the inventory level rises, due to the compressibility of natural gas.

The second model iteration changes the maximum daily injection and withdrawal constraints to be piecewise linear instead of constant. This introduced a unique challenge by itself, because piecewise linear functions have the ability to be non-convex, yet in this case the injection and withdrawal constraints are more realistic. The model was validated against the original simplified model and showed consistency. This granted the ability to model more complex logistic constraints while maintaining a similar degree of complexity as the original model.

The models described thus far have the primary advantage of being much less complex than existing models. For natural gas trading, this is advantageous due to the undesirability of "black-box" models, which traders may find to be either computationally or conceptually intractable. While the models are not able to incorporate all possible information, they provide a valuable means by which natural gas traders can shore up their own calculations.

9 Future Work

9.1 Options Trading

Expected completion: TBD

The current model uses forward contract or futures trading as a basis for profit calculation. However, many natural gas models instead involve options trading, where owners of options have the choice to exercise an option. Options trading in this form also tends to involve hedging against the option.

References

[1] Thompson, M., Davison, M. and Rasmussen, H. “Natural gas storage valuation and optimization: A real options application”. *Naval Research Logistics, 56* (2009): 226238.

[2] Boogert, A., C. De Jong. “Gas storage valuation using a Monte Carlo method”. *Journal of Derivatives 15* (2008): 8198.

[3] Bjerksund, Petter and Stensland, Gunnar and Vagstad, Frank. “Gas Storage Valuation: Price Modelling v. Optimization Methods” *NHH Dept. of Finance & Management Science Discussion* (2008): 2008/20.

[4] Li, Y. Natural Gas Storage Valuation. Master Thesis, Georgia Institute of Technology. *Managerial Finance 31* (2007): 7495.

[5] Bringedal, Bård. “Valuation of Gas Storage: A Real Options Approach, Masters Thesis”. *The Norwegian University of Science and Technology* (2003).

[6] Lai, G., F. Margot, N. Secomandi. “An approximate dynamic programming approach to benchmark practice-based heuristics for natural gas storage valuation”. *Operations Research 58* (2010a): 564582.

[7] Clewlow, L., C. Strickland. “A multi-factor model for energy derivatives”. *Lacima group publications* (1999).

[8] Guenin, B., J. Knemann, and Levent Tuncel. ”Nonlinear Optimization.” A Gentle Introduction to Optimization. *Cambridge: U Printing House* (2014).