Closed-Form for State (AD) Price Matrix

Written by Youngjin Jung
(yj2860@columbia.edu)

Summary

We develop a closed-form solution for recovering the state price matrix (Arrow-Debreu prices) from observed European option prices across multiple tenors. Building on the framework of Ross (2015), which models the economy as a finite, time-homogeneous Markov process, we address the key limitation of prior empirical methods – numerical instability – by deriving an analytical expression using vectorization and Kronecker products.

With the number of tenors n and the number of steps per a tenor k, the maximum number of states in the economy $N \coloneqq (n-1)k+1$. Referring the initial state as ω_r where $1 \le r \le N$, we assume that we observed spot state price vectors π_t with time to maturities $t = 1, \dots, nk$, which follow the induction relationship $\pi_{t+1} = Q\pi_t$ where Q is the common ratio matrix.

The algorithm is as follows:

1) Remove the r-th element of each π_t to construct reduced vector:

$$\pi'_t = \left[\pi_t^{r,1}, \dots, \pi_t^{r,r-1}, \pi_t^{r,r+1}, \dots, \pi_t^{r,n}\right]^T$$

2) Calculate of intermediate variables for calculation:

$$K_{(N-1)\times(N-1)} = [\pi'_1, \dots, \pi'_{N-1}]^T, \qquad B_{(N-1)\times N} = \left[\pi_{t=k+1} - \pi_{t=k}\pi_{t=1}^{r,r}, \dots, \pi_{t=nk} - \pi_{t=k}\pi_{t=(n-1)k}^{r,r}\right]$$

$$\left[q^{(1)}, \dots, q^{(r-1)}, q^{(r+1)}, \dots, q^{(N)}\right] = (K^{-1}B)^T, \qquad q^{(r)} = \pi'_k$$

3) Get the closed form of Q and the transition state price Π:

$$Q = \left[q^{(1)}, \dots, q^{(N)}\right], \qquad \Pi = Q^T$$

1. Introduction

1.1. List of Symbols

Symbol	Description
n	Number of monthly tenors (option maturities)
k	Number of time steps per month (e.g. weekly = 4, daily = 30)
N	Number of states: $N = (n-1)k + 1$
$\Omega = (\omega_1, \dots, \omega_N)$	State space of the economy (e.g. (bull, flat, bear market)
ω_r	Initial state of the economy (state r, $1 \le r \le N$)
$\pi^{(r,j)}$	(Unit period) State price (AD security price) for transition from state ω_r to state ω_j
$\Pi = \left[\pi^{(r,j)}\right]_{n\times m}$	Matrix of state prices
$\pi_{t=\tau}^{(r,j)}$	Spot state price for transition from state ω_r to state ω_j with time to maturity of $t = \tau$
$\pi_{t=\tau} = \left[\pi_{t=\tau}^{(r,1)}, \dots, \pi_{t=\tau}^{(r,N)}\right] \in \mathbb{R}^N$	Vector of spot state prices

1.2. Fundamental Framework

We start with a simple example – three possible states of the economy, for example S&P 500 index can only take states in Bull, Flat, and Bear Market.

$$\Omega = (\omega_1, \omega_2, \omega_3)$$

And, we intentionally choose the middle (second) state ω_2 (Flat market) as the initial state to aid understanding.

$$\text{current state} \coloneqq \omega_2$$

Additionally, we refer the current state with the index r.

Our target is to find the state price (AD price) matrix Π. For calculation, we impose a dummy variable:

$$Q \coloneqq \Pi^T = \begin{bmatrix} \pi^{1,1} & \pi^{2,1} & \pi^{3,1} \\ \pi^{1,2} & \pi^{2,2} & \pi^{3,2} \\ \pi^{1,3} & \pi^{2,3} & \pi^{3,3} \end{bmatrix} =: \left[q^{(1)}, q^{(2)}, q^{(3)}\right]$$

Jensen et al (2019) showed that the recovery is only possible when the number of states is not greater than the number of tenors we observed. So, in the following argument, we impose the same number of states and tenors.

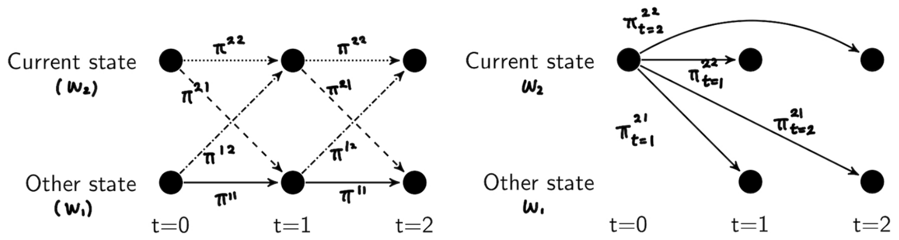
$$t = 1,2,3$$

For tenor (time to maturity, monthly) $t = 1,2,3$, we denote the spot state prices, which are derived from the option quotes data with SVI Interpolation and Local Volatility techniques, as:

$$\pi_1 \coloneqq \begin{bmatrix} \pi_1^{r,1} \\ \pi_1^{r,2} \\ \pi_1^{r,3} \end{bmatrix}, \qquad \pi_{t=2} \coloneqq \begin{bmatrix} \pi_2^{r,1} \\ \pi_2^{r,2} \\ \pi_2^{r,3} \end{bmatrix}, \qquad \pi_{t=3} \coloneqq \begin{bmatrix} \pi_3^{r,1} \\ \pi_3^{r,2} \\ \pi_3^{r,3} \end{bmatrix}$$

where $r = 2$.

For clarification, I have attached two figures. The lefthand side represents the time-homogeneous Markov process about state prices ($Q = \Pi^T$), and the righthand side represent that we can only observe the spot state prices with the initial state of ω_2 ($\pi_1, \pi_2, \dots$). We refer state prices (in an unit period) with the notation $\pi^{r,j}$ and "spot" state prices with subscript t, like $\pi_t^{r,j}$



Since the initial state is ω_2, the second column of the state price matrix is known and equivalent to $\pi_{t=1}$, and we aren't able to know the others at the initial state.

$$q^{(2)} = \pi_{t=1} \text{ (known)}, \qquad Q' := \left[q^{(1)}, q^{(3)}\right] \ (unknown)$$

Now, our problem reduces to find Q' using spot prices $\pi_{t=1}, \pi_{t=2}, \pi_{t=3}$ and the induction relationship such that $\pi_{t=k} = Q^{k-1}\pi_{t=k-1}$. Intuitively, we can guess the existence of the solutions because we have 6 equations from the spot state prices and 6 unknown variables in Q'.

$$\pi_{t=1} = q^{(2)} = \begin{bmatrix} \pi_{t=1}^{2,1} \\ \pi_{t=1}^{2,2} \\ \pi_{t=1}^{2,3} \end{bmatrix}, \qquad \pi_{t=2} = Q\pi_{t=1}, \qquad \pi_{t=3} = Q\pi_{t=2} = Q^2\pi_{t=1}$$

With the 6 equations and unknown variables, the matrix Q can be uniquely determined by solving the corresponding induction relationship.

1.3. Limitations of Existing Approaches

1.3.1. Ross (2015): The Recovery Theorem (cited 460)

Ross introduced a recovery framework, assuming a time-homogeneous, finite state Markov process under the physical measure, but just proposed very simple the coarse case of 12-by-12 grid that might over simplify the problem, amplifying estimation error of recovered physical probability density.

1.3.2. Audrino, Huitema, and Ludwig (2015): Empirical Analysis of Ross Recovery (cited 43)

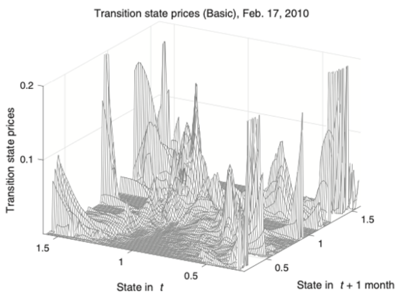
Estimated the smoothed state price curve in nonparametric way and then recover Q by enforcing consistency across overlapping maturities. The solution is obtained through numerical methods rather than a closed-form analytical expression.

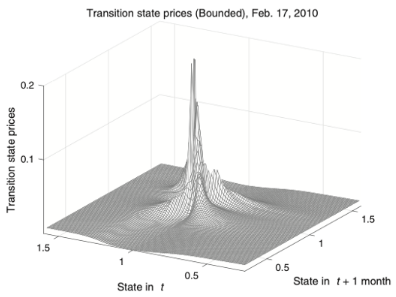
1.3.3. Audrino, Huitema, and Ludwig (2019): An empirical implementation of the Ross recovery theorem as a prediction device (cited 41)

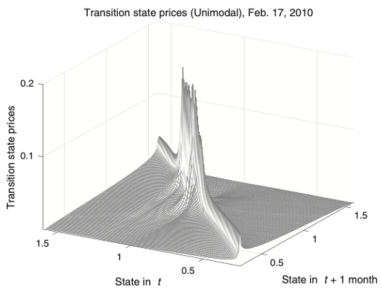
Proposed overlapping methods, but still opted numerical approach to find Q

1.3.4. Jackwerth and Menner (2020): Does the Recovery Theorem Work Empirically? (cited 77)

The estimation of Q is performed via optimization. However, as illustrated in the figure below, the results vary significantly on how the constraints in the objective function are formulated.







In these figures, they are referred as the following objective function of (1), (2), and (3) from left to right, and the z-axis represents recovered (transition) state prices $\pi^{r,j}$.

(1): $\min_{\pi^{i,j}} \sum_{j\in I}\sum_t\left(\pi_{t+10}^{r,j} - \sum \pi_t^{r,h}\pi^{h,j}\right)^2 \ s.t. \pi^{i,j} > 0$

(2): $\min_{\pi^{i,j}} \sum_{j\in I}\sum_t\left(\pi_{t+10}^{r,j} - \sum \pi_t^{r,h}\pi^{h,j}\right)^2 \ s.t. \pi^{i,j} > 0 \ and \ 0.9 \le \sum \pi^{i,j} \le 1.0 \ \forall i \in I$

(3): $\min_{\pi^{i,j}} \sum_{j\in I}\sum_t\left(\pi_{t+10}^{r,j} - \sum \pi_t^{r,h}\pi^{h,j}\right)^2 \ s.t. \pi^{i,j} > 0, 0.9 \le \sum \pi^{i,j} \le 1.0 \ \forall i \in I, \pi^{i,j_1} \le \pi^{i,l_1}, and \ \pi^{i,j_2} \ge \pi^{i,l_2}$

1.4. Overcoming the limitations by finding closed form

As demonstrated in previous studies, numerical methods for estimating the transition matrix Q tend to be highly unstable. To address this issue, we derive a mathematical closed form for Q. Specifically, we exploit vectorization and the Kronecker product to solve the system of induction equations:

$$\pi_{t+1} = Q\pi_t \ \ for \ t = 1, \dots, n$$

given spot state price vectors $\pi_1, \dots, \pi_n$.

2. Closed Form for Ross Original

Ross (2015) assumes homogeneous Markov process. This implies, as illustrated in 1.3., the spot state prices are generated through recursive application of the state price matrix over the N-state space. Formally, this relationship is expressed as follows:

$$\pi_{t+1}^{r,j} = \sum_{h=1}^{N} \pi_t^{r,h}\pi^{h,j} \ \ for \ \forall j = 1, \dots, N \ and \ t = 1, \dots, n$$

We introduce the mathematical solution finding the state price matrix $\Pi = Q^T$ given spot state price vectors π_t where $t = 1, \dots, n$.

2.1. Mathematical Tools

To solve the system of induction, we leverage vectorization and Kronecker tricks.

2.1.1. Vectorization Operator

This operator stretch a matrix out into a single vector:

$$vec\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \begin{bmatrix} a \\ c \\ b \\ d \end{bmatrix}, \qquad vec(A) = \left[a_{1,1}, \dots, a_{m,1}, a_{1,2}, \dots, a_{m,2}, \dots, a_{1,n}, \dots, a_{m,n}\right]^T$$

2.2.2. Kronecker Product (of $A_{m\times n} \otimes B_{p\times q}$)

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \dots & \dots & \dots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

,which is a $pm \times qn$ matrix.

2.2.3. Roth's Lemma

$$vec\ (ABC^T) = (C \otimes A) vec(B)$$

2.1. Illustrative Example

We now proceed to estimate the matrix Q using the illustrative setup introduced in Section 1.

For any vector $x = [x_1, x_2, x_3]^T$,

$$Qx = q^{(1)}x_1 + q^{(2)}x_2 + q^{(3)}x_3$$

The equation gives:

$$\begin{cases} \pi_{t=2} = Q\pi_{t=1} = q^{(1)}\pi_{t=1}^{2,1} + q^{(2)}\pi_{t=1}^{2,2} + q^{(3)}\pi_{t=1}^{2,3} \\ \pi_{t=3} = Q\pi_{t=2} = q^{(1)}\pi_{t=2}^{2,1} + q^{(2)}\pi_{t=2}^{2,2} + q^{(3)}\pi_{t=2}^{2,3} \end{cases}$$

This can be written in a vector form:

$$\begin{cases} \pi_{t=2} - q^{(2)}\pi_{t=1}^{2,2} = \left[q^{(1)}, q^{(3)}\right]\left[\pi_{t=1}^{2,1}, \pi_{t=1}^{2,3}\right]^T \\ \pi_{t=3} - q^{(2)}\pi_{t=2}^{2,2} = \left[q^{(1)}, q^{(3)}\right]\left[\pi_{t=2}^{2,1}, \pi_{t=2}^{2,3}\right]^T \end{cases}$$

Since $q^{(2)} = \pi_{t=1}$ and $Q' = \left[q^{(1)}, q^{(3)}\right]$,

$$\begin{cases} \pi_{t=2} - \pi_{t=1}\pi_{t=1}^{2,2} = Q'\left[\pi_{t=1}^{2,1}, \pi_{t=1}^{2,3}\right]^T \\ \pi_{t=3} - \pi_{t=1}\pi_{t=2}^{2,2} = Q'\left[\pi_{t=2}^{2,1}, \pi_{t=2}^{2,3}\right]^T \end{cases}$$

By the Roth's Lemma,

$$\begin{cases} vec\left(\pi_{t=2} - \pi_{t=1}\pi_{t=1}^{2,2}\right) = vec\left(Q'\left[\pi_{t=1}^{2,1}, \pi_{t=1}^{2,3}\right]^T\right) = \left(\left[\pi_{t=1}^{2,1}, \pi_{t=1}^{2,3}\right] \otimes I_3\right) vec(Q') \\ vec\left(\pi_{t=3} - \pi_{t=1}\pi_{t=2}^{2,2}\right) = vec\left(Q'\left[\pi_{t=2}^{2,1}, \pi_{t=2}^{2,3}\right]^T\right) = \left(\left[\pi_{t=2}^{2,1}, \pi_{t=2}^{2,3}\right] \otimes I_3\right) vec(Q') \end{cases}$$

Stacking those two equation horizontally gives:

$$\left(\begin{bmatrix} \pi_{t=1}^{2,1} & \pi_{t=1}^{2,3} \\ \pi_{t=2}^{2,1} & \pi_{t=2}^{2,3} \end{bmatrix} \otimes I_3\right) vec(Q') = \begin{bmatrix} \pi_{t=2} - \pi_{t=1}\pi_{t=1}^{2,2} \\ \pi_{t=3} - \pi_{t=1}\pi_{t=2}^{2,2} \end{bmatrix}$$

Let $K = \begin{bmatrix} \pi_{t=1}^{2,1} & \pi_{t=1}^{2,3} \\ \pi_{t=2}^{2,1} & \pi_{t=2}^{2,3} \end{bmatrix}$, $b = \begin{bmatrix} \pi_{t=2} - \pi_{t=1}\pi_{t=1}^{2,2} \\ \pi_{t=3} - \pi_{t=1}\pi_{t=2}^{2,2} \end{bmatrix}$ and $M = K \otimes I_3$, if there the K is invertible, we can find Q' by:

$$vec({Q'}^*) = \begin{bmatrix} {q^{(1)}}^* \\ {q^{(3)}}^* \end{bmatrix} = M^{-1}b = (K^{-1} \otimes I_3)b$$

Therefore, we can find the state price matrix Q and Π by:

$$Q^* = \left[q^{(1)^*}, \pi_{t=1}, q^{(3)^*}\right], \quad and \quad \Pi^* = Q^{*T}$$

2.2. Closed Form

Suppose that spot state prices are observed at each time step $t = 1, \dots, n$. Since, on basic Ross setting, the maximum number of possible states in the economy is $N = n$, spot state prices are represented by:

$$\pi_t = \left[\pi_t^{r,1}, \dots, \pi_t^{r,n}\right]^T$$

where $\pi_t^{(r,s)}$ refers the price at time t of a state price (an Arrow-Debreu security) that pays $1 if the state at the maturity t is ω_s given the current state is ω_r.

We set the initial state as w_r, where $1 \leq r \leq N$.

For convenience of calculation, we introduce dummy matrix Q and vector $q^{(i)}$:

$$Q \coloneqq \Pi^T = \begin{bmatrix} \pi^{1,1} & \dots & \pi^{N,1} \\ \dots & \dots & \dots \\ \pi^{1,N} & \dots & \pi^{N,N} \end{bmatrix} =: \left[q^{(1)}, \dots, q^{(N)}\right]$$

As we discussed before, the state price matrix satisfies the following induction structure:

$$Q' = \left[q^{(1)^*}, \dots, q^{(r-1)^*}, q^{(r+1)^*}, \dots, q^{(n)^*}\right]$$

$$\pi'_t = \left[\pi_t^{r,1}, \dots, \pi_t^{r,r-1}, \pi_t^{r,r+1}, \dots, \pi_t^{r,n}\right]^T$$

Given π_t for $t = 1,2, \dots, n$,

$$\begin{cases} \pi_2 = Q\pi_1 = q^{(1)}\pi_1^{r,1} + \dots + q^{(r)}\pi_1^{r,r} + \dots + q^{(n)}\pi_1^{r,n} \\ \dots \\ \pi_n = Q\pi_{n-1} = q^{(1)}\pi_{n-1}^{r,1} + \dots + q^{(r)}\pi_{n-1}^{r,r} + \dots + q^{(n)}\pi_{n-1}^{r,n} \end{cases}$$

Since $q^{(r)} = \pi_1$, we get:

$$\begin{cases} \pi_2 - \pi_1\pi_1^{r,r} = q^{(1)}\pi_1^{r,1} + \dots + q^{(r-1)}\pi_1^{r,r-1} + q^{(r+1)}\pi_1^{r,r+1} + \dots + q^{(N)}\pi_1^{r,n} \\ \dots \\ \pi_n - \pi_1\pi_{n-1}^{r,r} = q^{(1)}\pi_{n-1}^{r,1} + \dots + q^{(r-1)}\pi_{n-1}^{r,r-1} + q^{(r+1)}\pi_{n-1}^{r,r+1} + \dots + q^{(N)}\pi_{n-1}^{r,n} \end{cases}$$

Summarize with matrix notations gives:

$$\begin{cases} \pi_2 - \pi_1\pi_1^{r,r} = Q'\pi'_1 \\ \dots \\ \pi_n - \pi_1\pi_{n-1}^{r,r} = Q'\pi'_{n-1} \end{cases}$$

Applying Roth's Lemma, we vectorize this equation to obtain:

$$\begin{cases} vec(\pi_2 - \pi_1\pi_1^{r,r}) = vec(Q'\pi'_1) = \left(\pi'^T_1 \otimes I_n\right)vec(Q') \\ \dots \\ vec(\pi_n - \pi_1\pi_{n-1}^{r,r}) = vec(Q'\pi'_{n-1}) = \left(\pi'^T_{n-1} \otimes I_n\right)vec(Q') \end{cases}$$

By stacking the system, we get:

$$b = Mvec(Q') = (K \otimes I_n)\, vec(Q')$$

where $K = [\pi_1', \dots, \pi_{n-1}']^T$, $b = vec([\pi_2 - \pi_1 \pi_1^{r,r}, \dots, \pi_n - \pi_1 \pi_{n-1}^{r,r}])$, and $M = (K \otimes I_n)$.

Thus, the solution of Q' is:

$$vec(Q'^{*}) = \left[q^{(1)^*}, \dots, q^{(r-1)^*}, q^{(r+1)^*}, \dots, q^{(N)^*}\right] = M^{-1}b = (K^{-1} \otimes I_n)b$$

In conclusion, we can retrieve

$$\Pi = Q^T = \left[q^{(1)^*}, \dots, q^{(r-1)^*}, \pi_1, q^{(r+1)^*}, \dots, q^{(N)^*}\right]^T$$

2.3. Validation with Simulation

1) Randomly generate a sequence of spot state prices $\pi_1, \dots, \pi_n$
2) Estimate the transition matrix Q using the algorithm described above.
3) Construct the estimated spot prices $\hat{\pi}_t$ recursively using the induction relationship: $\hat{\pi}_t = Q\hat{\pi}_{t-1}$
4) Check whether the estimated prices match the original simulated prices: $\hat{\pi}_t \approx \pi_t$

2.4. Implementation

The reader can find the code in the github link in Appendix

```
----------------------------------------------------------------------------------------------------
n (# of time points observed): 120
N (# of possible states): 120
----------------------------------------------------------------------------------------------------
Initial state: w6
----------------------------------------------------------------------------------------------------
All Tests Passed
Time Required: 1.8267 seconds
Pi: (120, 120)
----------------------------------------------------------------------------------------------------
```

3. Overlapping Method

We follow the overlapping method framework outlined by Audrino et al. (2020). As discussed in Jensen et al. (2019), the number of observed option tenors must be at least as large as the number of economic states in order to identify the state prices. While our earlier setup used monthly intervals as the minimum time unit, we now construct a finer time grid by dividing the period into a smaller interval, such as weeks or days. This generalizes the prior method, Ross Basic.

3.1. Closed Form

We denote by n the number of tenors and by k the number of steps per a tenor. For instance, if we consider a three-month horizon with weekly frequency, then $n = 3$ and $k = 4$. Matching the number of equations and unknown variables, the maximum total number of states in the economy is given by:

$$N = (n-1)k + 1$$

We refer to the initial state as w_r, where $1 \leq r \leq N$.

We suppose that spot state prices are observed at each time step $t = 1, \dots, nk$, denoted as:

$$\pi_t = \left[\pi_t^{r,1}, \dots, \pi_t^{r,N}\right]^T$$

where $t = 1, \dots, N$ and $\pi_t^{(r,s)}$ represents the price at time t of a state price (an Arrow-Debreu security) that pays \$1 if the state at the maturity t is ω_s given the current state is ω_r.

As we discussed before, the state price matrix satisfies the following induction structure:

$$\pi_t = Q\pi_{t+1} \quad or \quad \pi_{t+s} = Q^s \pi_t \quad for \quad t = 1, \dots, nk$$

$$and \quad \pi_k = \left[\pi_k^{r,1}, \dots, \pi_k^{r,N}\right]^T = q^{(r)}$$

We use the same setting in Section 2 as:

$$Q \coloneqq \Pi^T = \begin{bmatrix} \pi^{1,1} & \dots & \pi^{N,1} \\ \dots & \dots & \dots \\ \pi^{1,N} & \dots & \pi^{N,N} \end{bmatrix} =: \left[q^{(1)}, \dots, q^{(N)}\right]$$

$$Q' = \left[{q^{(1)}}^*, \dots, {q^{(r-1)}}^*, {q^{(r+1)}}^*, \dots, {q^{(N)}}^*\right]$$

$$\pi'_t = \left[\pi_t^{r,1}, \dots, \pi_t^{r,r-1}, \pi_t^{r,r+1}, \dots, \pi_t^{r,n}\right]^T$$

In the same way in Section 2,

$$vec(Q'^*) = \left[{q^{(1)}}^*, \dots, {q^{(r-1)}}^*, {q^{(r+1)}}^*, \dots, {q^{(N)}}^*\right] = M^{-1}b = (K^{-1} \otimes I_N)b$$

where $K = [\pi'_1, \dots, \pi'_{N-1}]^T$, and $b = vec([\pi_{k+1} - \pi_k \pi_1^{r,r}, \dots, \pi_N - \pi_k \pi_{N-k}^{r,r}])$

Thus, the solution of state price matrix Π given spot state prices $\pi_1, \dots, \pi_{nk}$ is:

$$\Pi = Q^T = \left[{q^{(1)}}^*, \dots, {q^{(r-1)}}^*, \pi_{t=k}, {q^{(r+1)}}^*, \dots, {q^{(N)}}^*\right]^T$$

3.2. Implementation

You can find the implementation in the github link in Appendix.

```
-----------------------------------------------------------------------------------------------
# of time points observed: 120 (12 months & 10 steps in a month)
# of possible states: 111
-----------------------------------------------------------------------------------------------
All Tests Passed
Time Required: 38.1369 seconds
Pi: (111, 111)
```

3.3. Computational Limitations

Since the size of the matrix M is $N(N-1) \times N(N-1)$, solving the system via direct inversion or LU decomposition requires (N^6) time complexity. Even for a relatively simple case, such as 12 months with 4 time steps (weekly) per month, the number of states N becomes 45, resulting in roughly 10^{10} operations. So, we can only cover a limited number of states due to the computational infeasibility, which makes this approach useless. We have solved the issue in the following section.

4. Accelerated Overlapping Method

To address the time complexity problem, we leverage the structure of the Kronecker product and reformulate the system using matrix equations instead of vectorized forms.

4.1. Closed Form

We recognize the equivalence:

$$vec(Q') = (K^{-1} \otimes I_N)b \;\Leftrightarrow\; KQ'^T = B$$

where $B_{N-1\times N} = \left[\pi_{t=k+1} - \pi_{t=k}\pi^{r,r}_{t=1},\ \dots\ ,\ \pi_{t=nk} - \pi_{t=k}\pi^{r,r}_{t=(n-1)k}\right]$, that is $vec(B) = b$

This reformulation significantly reduces the time complexity from $O(N^6)$ to $O(N^3)$. This complexity reduction allows us to handle quadratically as many states under the same computational resources. For example, if we handle just 45 states (12 months and 4 time steps) before, now we can address 2025 states. This makes it practical to handle almost any number of state spaces observed in the option market.

4.2. Comparison Analysis:

----

n (# of months): 36 (3.0 years)
k (# of steps in a month): 4
*n * k (# of time points observed): 144*
N (# of possible states): 141

Initial state: w5

Basic Overlapping:

All Tests Passed
Time Required: 208.3795 seconds
Pi: (141, 141)

Accelerated Overlapping:

All Tests Passed

Time Required: 0.1121 seconds

Pi: (141, 141)

----

5. Concrete Example

----

n (# of months): 60 (5.0 years)

k (# of steps in a month): 4 (weekly)

*n * k (# of time points observed): 240*

N (# of possible states): 237

We can partition the total state space into a granular grid of 237 points

i.e.) [-5.0σ, -4.958σ, -4.915σ, ... , 4.915σ, 4.958σ, 5.0σ]

Assume that the initial state is w_100 = -0.781σ

Suppose that we observed weekly spot state prices from SPXW option data as:

Today is Jan 1, 2025

(week 1) SPXW 06/01/25: [0.18, 0.6, '...', 0.72, 0.18]

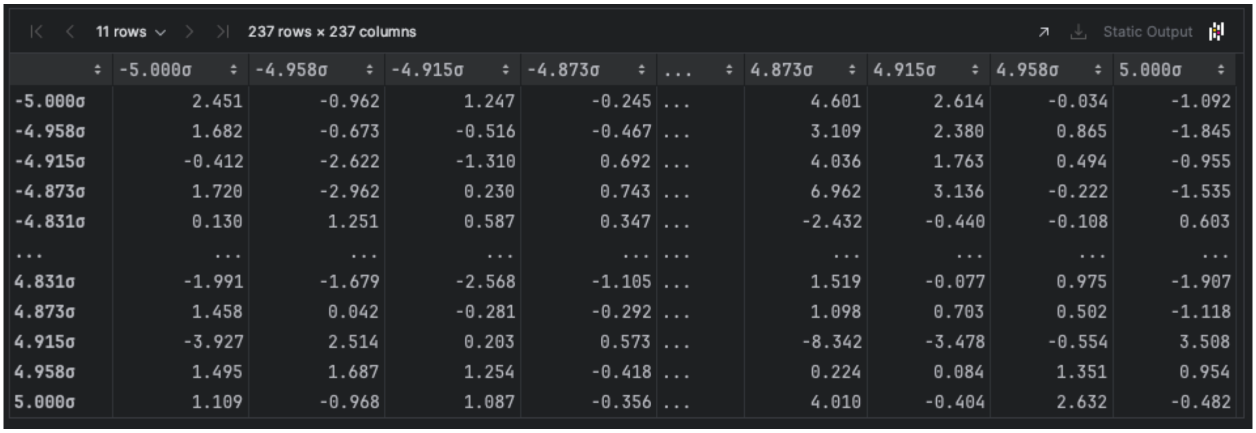
(week 0) SPXW 06/08/25: [0.44, 0.49, '...', 0.91, 0.51]

...

(week 239) SPXW 12/23/29: [0.33, 0.3, '...', 0.32, 0.39]

(week 240) SPXW 12/30/29: [0.46, 0.92, '...', 0.52, 0.32]

The transition state price matrix recovered from the spot state prices is:



	-5.000σ	-4.958σ	-4.915σ	-4.873σ	...	4.873σ	4.915σ	4.958σ	5.000σ
-5.000σ	2.451	-0.962	1.247	-0.245	...	4.601	2.614	-0.034	-1.092
-4.958σ	1.682	-0.673	-0.516	-0.467	...	3.109	2.380	0.865	-1.845
-4.915σ	-0.412	-2.622	-1.310	0.692	...	4.036	1.763	0.494	-0.955
-4.873σ	1.720	-2.962	0.230	0.743	...	6.962	3.136	-0.222	-1.535
-4.831σ	0.130	1.251	0.587	0.347	...	-2.432	-0.440	-0.108	0.603
...	...	...	...	...	...	...	...	...	...
4.831σ	-1.991	-1.679	-2.568	-1.105	...	1.519	-0.077	0.975	-1.907
4.873σ	1.458	0.042	-0.281	-0.292	...	1.098	0.703	0.502	-1.118
4.915σ	-3.927	2.514	0.203	0.573	...	-8.342	-3.478	-0.554	3.508
4.958σ	1.495	1.687	1.254	-0.418	...	0.224	0.084	1.351	0.954
5.000σ	1.109	-0.968	1.087	-0.356	...	4.010	-0.404	2.632	-0.482

Appendix

Github link: