Pricing Stock Options under the Heston Stochastic-Volatility Model via a Trinomial Tree in a Regime-Switching Economy with added transaction costs

Alan Berdikulov

First reader: Dr. Dongming Wei
Second reader: Dr. Piotr Sebastian Skrzypacz

April 2025

Abstract

In this paper we will discuss the procedure on how to develop computationally efficient trinomial-tree method for pricing only for European call options under Heston's stochastic-volatility model, incorporating both regime-switching approximations and proportional transaction costs. A decorrelation transform developed by Beliaeva and Nawalkha [4] and discretization of the variance process via a continuous-time Markov chain is applied to the Heston model, yielding the reduction of the two-dimensional dynamics to a family of coupled one-dimensional processes. On each regime, we build a recombining trinomial lattice whose jump-sizes match the local drift and variance, and we connect the lattices across regimes using the Markov-chain generator. This yields a five-point tree with only $O(N)$ nodes at step N, in contrast to the exponential growth of naïve multinomial schemes [11]. At each section the paper will be as detailed as possible when deriving the PDE variations starting from the famous Heston model [5] ending with the option's PDE which is dependent on Markov Chain generator. Numerical approximations show that this method achieves less than 1% accuracy on benchmark European calls only for a few dozen variance states and a few thousands time-steps. We further include proportional transaction costs to our grid design of the trinomial tree by penalizing deltas (hedge) at each node, and we demonstrate the realizations on the same example as with plain trinomial tree.

1 Introduction

Although the Black Scholes model [1] was at the forefront of the option pricing, its relaxed assumption of constant volatility usually led to the inaccurate pricing results when they were compared to the observed market data which often has the phenomena as "volatility smile" and "skew" as discussed in Jankova (2018) [9]. To get rid of this issue the ideas of stochastic volatility models were proposed by numerous authors including Hillard & Schwarz [**?**] who gave accurate depiction on the dynamics of stochastic-volatility models. Amongst them Heston [5] had the most renowned one. Those stochastic volatility models [6] incorporate randomness in volatility making the option pricing model much harder. In spite of the novelty of those models they are complex and they have almost no analytical solutions, especially when we consider path-dependent options. To address this issue numerical methods such as tree-based methods are often used for approximation Tree-based methods are of particular interest due to their simplicity and providing the intuition for financial interpretation. They get the stochastic differential equations underlying assets discretized, therefore improving their efficiency by recombining property, which limits the exponential growth of tree branches with the increase of the time steps. Cox-Ross-Rubinstein [3] proposed the idea of Binomial Option Pricing model which used to approximate Leland's model [10]. They showed that if the time step size goes to zero, their binomial model converges to Black-Scholes price. Taking the limit and squeezing the time steps, the Binomial Option pricing model [3] converges to the BS model. Whilst tree methods may perform better for constant volatility models such as BS model, stochastic volatility models such as Heston model [5] have varying volatility which makes it difficult for the recombination of tree branches and therefore having low computational efficiency. Lots of methods have been developed to address this problem such as Nelson-Ramaswamy transformation [12], which transforms the volatility terms constant. Nevertheless, these methods often lead to unprecedented issues such as joint probabilities being negative or the tree structures being inefficient. Liu proposed an idea of regime-switching model [11] with a recombining tree which subsequently improves the efficiency that still requires a big number of branches at each time step. His method of discretization of the volatility that attain different values for each of the state has gained big momentum lately. It allows to accuratelty approximate the heston stochastic-volatility, with each more better refinement converges his finite-difference method to the heston PDE. We will focus on Liu's regime-switching model and its application of making the volatility process conditionally independent. His approach still requires a lot of branching at each time step. We will explore the Liu approach applied to a trinomial tree structure that will reduces the branch numbers for each time increment that is manifested in Zeng's paper [13]. Then we will use the results of Zeng and Zhu [13] to note the similarity between the trinomial tree's and the finite difference approach for the Heston model. We also should account for the transaction costs, so the idea of proportional transaction costs offered by Boyle-Vorst will be used to enhance the model to have more real life application [2], for each of this section providing detailed explanations for the derivations. As a result, I will give the python code for this method of valuing the stock options and its variation with transaction costs

2 Heston's Stochastic Volatility Model

In the Heston model, the volatility σ_t follows a mean-reverting process:

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dB_t^1 \tag{1}$$

$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t} dB_t^2 \tag{2}$$

where B_t^1 and B_t^2 are correlated Brownian motions 2, with ρ being the correlation coefficient.

$$dB_t^2 = \rho dB_t^1 + \sqrt{1-\rho^2} dW_t \tag{3}$$

where W_t is a Brownian dependent on B_t^1 . To derive the HEston PDE, we proceed as follows:

Form a portfolio consisting of on option $V = V(S.v,t)$, Δ amounts of the underlying stock and β amount of other option $U = U(S,v,t)$ Therefore our portfolio will have the following value

$$\Pi = V(S,v,t) + \Delta S + \beta U(S,v,t)$$

Portfolio value Π also time dependent so $\Pi = \Pi_t$. Using the self-financing argument, an argument where investor rebalances his portfolio so that at any given time his portfolio has no risk, the change in portfolio becomes:

$$d\Pi = dV(S,v,t) + \Delta dS + \beta dU(S,v,t)$$

Then, we could apply Ito's Lemma on our original option V, by differentiating w.r.t to v, t, S

$$\begin{aligned} dV &= \frac{\partial V}{\partial t}\, dt + \frac{\partial V}{\partial S}\, dS + \frac{\partial V}{\partial v}\, dv \\ &\quad + \frac{1}{2}\, vS^2 \frac{\partial^2 V}{\partial S^2}\, dt + \frac{1}{2}\, \sigma^2 v \frac{\partial^2 V}{\partial v^2}\, dt + \sigma\, v\, \rho\, S \frac{\partial^2 V}{\partial v\, \partial S}\, dt \end{aligned}$$

Applying Ito's Lemma on second option U, we get the same result. We could combine Ito representations of V and U with the change in portfolio to arrive at the following result:

$$\begin{aligned} d\Pi &= dV + \Delta\, dS + \phi\, dU \\ &= \left\{ \frac{\partial V}{\partial t} + \tfrac{1}{2}\, vS^2 \frac{\partial^2 V}{\partial S^2} + \rho\sigma v S \frac{\partial^2 V}{\partial v\, \partial S} + \tfrac{1}{2}\, \sigma^2 v \frac{\partial^2 V}{\partial v^2} \right\} dt \\ &\quad + \beta \left\{ \frac{\partial U}{\partial t} + \tfrac{1}{2}\, vS^2 \frac{\partial^2 U}{\partial S^2} + \rho\sigma v S \frac{\partial^2 U}{\partial v\, \partial S} + \tfrac{1}{2}\, \sigma^2 v \frac{\partial^2 U}{\partial v^2} \right\} dt \\ &\quad + \left\{ \frac{\partial V}{\partial S} + \beta \frac{\partial U}{\partial S} + \Delta \right\} dS + \left\{ \frac{\partial V}{\partial v} + \beta \frac{\partial U}{\partial v} \right\} dv\,. \end{aligned}$$

As we have mentioned earlier, Investor rebalances his portfolio at any given time to arrive at riskless portfolio, which means terms with S and v in the SDE that models Portfolio must be equal to zero. This results in the following hedge parameters.

$$\beta = -\frac{\frac{\partial V}{\partial v}}{\frac{\partial U}{\partial v}}, \quad \Delta = -\,\beta\,\frac{\partial U}{\partial S} - \frac{\partial V}{\partial S}.$$

Since we made our portfolio riskless, it must earn riskless interest rate r which makes our portfolio $d\Pi = r\Pi dt$. Exploiting the risklessness of our Portfolio we arrive at the following representation:

$$\begin{aligned} d\Pi &= dV + \Delta\, dS + \phi\, dU \\ &= \Big\{ \frac{\partial V}{\partial t} + \tfrac{1}{2}\, vS^2 \frac{\partial^2 V}{\partial S^2} + \rho\sigma vS \frac{\partial^2 V}{\partial v\, \partial S} + \tfrac{1}{2}\, \sigma^2 v \frac{\partial^2 V}{\partial v^2} \Big\} dt \\ &\quad + \,\beta \Big\{ \frac{\partial U}{\partial t} + \tfrac{1}{2}\, vS^2 \frac{\partial^2 U}{\partial S^2} + \rho\sigma vS \frac{\partial^2 U}{\partial v\, \partial S} + \tfrac{1}{2}\, \sigma^2 v \frac{\partial^2 U}{\partial v^2} \Big\} dt \end{aligned}$$

Let the differenital in terms of t be $[\dots]dt = Adt$ and differential in terms of S be $\beta[\dots]dS = \beta BdS$ making $d\Pi = (A + \beta B)dt$. Therefore $(A + \beta B) = r(V + \Delta S + \beta U)$. Substitute for β (hedging argument) we arrive at the following representation.

$$\frac{B - r\,U + r\,S\frac{\partial U}{\partial S}}{\frac{\partial U}{\partial v}} = \frac{A - r\,V + r\,S\frac{\partial V}{\partial S}}{\frac{\partial V}{\partial v}}$$

Left and Right hand sides of these two equations are very similar, implying that they could be written in terms of some function $f(S, v, t)$. According to Heston (1993) [5], this function is given by $f(S, v, t)$ by changing the phyisical measure in 2 to an arbitrary equivalent measure $\mathbf{Q}$ according to Girsanov's Theorem $dW_t^2 = dB_t^2 + \frac{\lambda(S_t, v_t, t)}{\sigma} dt$ yielding $dv_t = [k(\theta - v_t) - \lambda(S_t, v_t, t)]dt + dW_t^2$ where $\lambda(S_t, v_t, t)$ is a market risk, hence $f(S, v, t) = -k(\theta - v) + \lambda(S, v, t)$, where $\lambda(S, v, t)$ is a volatility risk. But, by initial construction, our portfolio is rebalanced at each time step making our portfolio riskless, so $\lambda(S, v, t) = 0$. Inserting our $f(S, v, t) = -k(\theta - v)$ into the right side and B we get famous Heston PDE, analytical solution of which will not be discussed in this Capstone Project:

$$\frac{\partial U}{\partial t} + \frac{1}{2} vS^2 \frac{\partial^2 U}{\partial S^2} + \rho\sigma Sv \frac{\partial^2 U}{\partial S \partial v} + \frac{1}{2}\sigma v^2 \frac{\partial^2 U}{\partial v^2} + rS\frac{\partial U}{\partial S} + \kappa(\theta - v)\frac{\partial U}{\partial v} - rU = 0 \tag{4}$$

where $U(S, v, t)$ is the option price. To solve it in European case, $U(S, v, t) = max(S - E, 0)$ where K is the strike price

3 Regime-Switching Stochastic Volatility Models

Liu [11] proposes the discretization of non-constant volatility via Markov chain, e.g. at each different regimes (states) they attain certain value. Assume P_t is a continuous-time Markov chain with K states. The state space is given by $\mathcal{P} := \{p_1, p_2, \dots, p_K\}$. The generator of the Markov chain P_t is denoted by $Q = (q_{ij})_{m\times n}$, which satisfies the following properties:

(i) $q_{ij} < 0$ for all $i \neq j$;

(ii) $\sum_{j\neq i} q_{ij} = 0$ for all $i = 1, 2, \dots, K$.

Then in a regime-switching model, the volatility term σ_t in (2.1) is a function of P_t:

$$\hat{\sigma}(P_t) = \begin{cases} \hat{\sigma}_1, & P_t = p_1, \\ \hat{\sigma}_2, & P_t = p_2, \\ \vdots & \\ \hat{\sigma}_K, & P_t = p_K. \end{cases}$$

Here the $\hat{\sigma}_i$ for $i = 1, 2, \dots, K$ are constant numbers. Thus the regime-switching SDE can be written as:

$$dS_t = \mu(P_t)S_t dt + \hat{\sigma}(P_t)S_t dB_t^1. \tag{5}$$

P_t is independent of the B_t^1. Since we have Markov chain, the PDE of a regime-switching model transforms into a PDE system for each of the regime. Solutions of this system must be solved simultaneously.

Suppose that an option of of some asset following the regime-switching economy 5 is priced at $U(S, P, t)$. It should be noted that $U(S, P, t)$ is a vector having following expression: $U(S, P, t) = (U(S, p_1, t), U(S, p_2, t), \dots, U(S, p_K, t))$. We then denote the i-th component of the solution vector, $U(S, p_i, t)$, by $U_i(S, t)$, i.e., for $i = 1, 2, \dots, K$. $U(S, P, t)$ is the solution of this PDE system for each i:

$$\frac{\partial U_i}{\partial t} + rS\frac{\partial U_i}{\partial S} + \frac{1}{2}\sigma_1^2 S^2 \frac{\partial^2 U_i}{\partial S^2} + rS\frac{\partial U_i}{\partial S} - rU_i + \sum_{j=1}^{K} q_{1j} U_j = 0, \tag{6}$$

4 Constructing the regime-switching model with proper inputs

According to Liu [11], the regime-switching model is a discretized version of the Heston model. Consider some stock whose price moves in accordance with 1 and

2 We will use the transformation Beliaeva and Nawalkha [4] since the regime switching model for each discrete volatility is independent from the stochastic process that models the stock's movement. It will allow us to decouple the two processes. Define the log-transformed variable X_t and the transformed variance process w_t with:

$$X_t = \ln\left(\frac{S_t}{S_0}\right) - \frac{\rho}{\sigma}(v_t - v_0) - \left(r - \frac{\rho\kappa\theta}{\sigma}\right) t \tag{7}$$

$$w_t = 2\sqrt{v_t} \tag{8}$$

By Itô's lemma for $X(S_t, v_t, t)$:

$$\begin{aligned} dX_t &= \frac{\partial X}{\partial t}\,dt + \frac{\partial X}{\partial S}\,dS_t + \frac{\partial X}{\partial v}\,dv_t \\ &\quad + \tfrac{1}{2}\,\frac{\partial^2 X}{\partial S^2}\,(dS_t)^2 + \tfrac{1}{2}\,\frac{\partial^2 X}{\partial v^2}\,(dv_t)^2 + \frac{\partial^2 X}{\partial S\,\partial v}\,(dS_t)(dv_t). \end{aligned}$$

Compute the partial derivatives:

$$\frac{\partial X}{\partial t} = -\left(r - \tfrac{\rho\kappa\theta}{\sigma}\right), \quad \frac{\partial X}{\partial S} = \frac{1}{S_t}, \quad \frac{\partial X}{\partial v} = -\frac{\rho}{\sigma},$$

$$\frac{\partial^2 X}{\partial S^2} = -\frac{1}{S_t^2}, \quad \frac{\partial^2 X}{\partial v^2} = 0, \quad \frac{\partial^2 X}{\partial S\,\partial v} = 0.$$

Using $(dS_t)^2 = v_t S_t^2\,dt$, $(dv_t)^2 = \sigma^2 v_t\,dt$, and $(dS_t)(dv_t) = \rho\,\sigma\,v_t S_t\,dt$, we substitute:

$$\begin{aligned} dX_t &= \left[-\left(r - \tfrac{\rho\kappa\theta}{\sigma}\right) + r - \frac{\rho}{\sigma}\,\kappa(\theta - v_t) - \tfrac{1}{2}\,v_t\right]dt + \frac{1}{S_t}\left(rS_t\,dt + \sqrt{v_t}\,S_t\,dB_t^1\right) \\ &\quad + \left(-\tfrac{\rho}{\sigma}\right)\left(\kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}\,dB_t^2\right) + \tfrac{1}{2}\left(-\tfrac{1}{S_t^2}\right)\left(v_t S_t^2\,dt\right) \\ &\quad + 0\cdot(\sigma^2 v_t\,dt) + 0\cdot(\rho\sigma v_t S_t\,dt) \\ &= \left(\tfrac{\rho\kappa}{\sigma} - \tfrac{1}{2}\right)v_t\,dt + \sqrt{v_t}\,dB_t^1 - \rho\,\sqrt{v_t}\,dB_t^2. \end{aligned}$$

Now set $v_t = \frac{1}{4}w_t^2$ and define $\alpha = \frac{\rho\kappa}{\sigma} - \frac{1}{2}$. Then

$$dX_t = \frac{1}{4}\,\alpha\,w_t^2\,dt + \sqrt{\tfrac{1}{4}w_t^2}\,\left(dB_t^1 - \rho\,dB_t^2\right) = \frac{1}{4}\,\alpha\,w_t^2\,dt + \tfrac{1}{2}\,w_t\big(dB_t^1 - \rho\,dB_t^2\big).$$

Finally introduce the independent Brownian motion

$$dW_t = \frac{dB_t^1 - \rho\,dB_t^2}{\sqrt{1-\rho^2}}, \quad \Longrightarrow \quad dB_t^1 - \rho\,dB_t^2 = \sqrt{1-\rho^2}\,dW_t,$$

to obtain the decoupled form

$$\boxed{dX_t = \frac{1}{4}\,\alpha\,w_t^2\,dt + \frac{1}{2}\,\sqrt{1-\rho^2}\,w_t\,dW_t.}$$

Let

$$w_t \;=\; 2\sqrt{v_t}.$$

Then

$$\frac{\partial w}{\partial v} = \frac{1}{\sqrt{v}}, \qquad \frac{\partial^2 w}{\partial v^2} = -\frac{1}{2\,v^{3/2}}.$$

By Itô's lemma

$$dw_t = w_v\,dv_t + \tfrac{1}{2}\,w_{vv}\,(dv_t)^2.$$

Since under the risk-neutral measure

$$dv_t = \kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}\,dB_t^2, \qquad (dv_t)^2 = \sigma^2\,v_t\,dt,$$

we substitute:

$$\begin{aligned} dw_t &= \frac{1}{\sqrt{v_t}}\Big[\kappa(\theta - v_t)\,dt + \sigma\sqrt{v_t}\,dB_t^2\Big] + \tfrac{1}{2}\Big(-\frac{1}{2\,v_t^{3/2}}\Big)\big[\sigma^2\,v_t\,dt\big] \\ &= \Big[\tfrac{2\kappa\theta - \frac{1}{2}\sigma^2}{2\sqrt{v_t}} - \frac{\kappa\sqrt{v_t}}{2}\Big]dt + \sigma\,dB_t^2. \end{aligned}$$

Noting $w_t = 2\sqrt{v_t}$, write

$$\phi(w) \;=\; \frac{2\kappa\theta - \frac{1}{2}\sigma^2}{w} \;-\; \frac{\kappa\,w}{2},$$

so that

$$\boxed{dw_t = \phi(w_t)\,dt \;+\; \sigma\,dB_t^2.}$$

We get the following system of conditionally independent SDE:

$$dX_t = \frac{1}{4}\alpha w_t^2 dt + \frac{1}{2}\sqrt{1-\rho^2}w_t dW_t \tag{9}$$

$$dw_t = \phi(w)dt + \sigma dB_t^2 \tag{10}$$

where

$$\alpha = (\frac{\rho\kappa}{\sigma} - \frac{1}{2}) \qquad W_t = \frac{B_t^1 - \rho B_t^2}{\sqrt{1-\rho^2}} \qquad \phi(w) = (2\kappa\theta - \frac{\sigma^2}{2})\frac{1}{w} - \frac{\kappa}{2}w$$

4.1 Discretization with Markov Chain Approximation

Follwoing Liu's construction [11] Define a Markov chain with N discrete states P_t that approximates the transformed variance process w_t, where:

$$P_t = \begin{cases} w_1, & P_t = 1, \\ w_2, & P_t = 2, \\ \vdots & \\ w_N, & P_t = N. \end{cases}$$

The drift and volatility terms in X_t are discretized accordingly:

$$dX_t = a(P_t)dt + b(P_t)dW_t, \tag{11}$$

Here X_t follows Geometric Brownian Motion where:

$$a(P_t) = \begin{cases} \frac{1}{4}w_1, & P_t = 1, \\ \frac{1}{4}w_2, & P_t = 2, \\ \vdots & \\ \frac{1}{4}w_N, & P_t = N. \end{cases} \qquad b(P_t) = \begin{cases} \frac{1}{2}\sqrt{1-\rho^2 w_1}, & P_t = 1, \\ \frac{1}{2}\sqrt{1-\rho^2 w_2}, & P_t = 2, \\ \vdots & \\ \frac{1}{2}\sqrt{1-\rho^2 w_N}, & P_t = N. \end{cases}$$

First, we have to make these two discretized functions coincide with the original continuous drift and volatility as the number of states shoots to the infinity. To do so, we let $w_n = n\Delta w$. Now we have to find Q-matrix generator of the Markov Chain. Liu does that by comparing coefficients of the PDE of 11 and 10

4.2 Finite Difference Approximation for PDE following SDE and regime-switching economy

Let us present the PDE of the system of stochastic differential equations in 10. Consider the pricing PDE for an option with maturity T and let $H(x, v, t)$ be the solution of this PDE following risk-neutral probability:

Let $H(x, w, t)$ denote the undiscounted price, at state $(X_t = x, w_t = w)$ and time $t \le T$, of an option paying $H(X_T, w_T, T)$ at maturity T. Applying Itô's lemma to $H(X_t, w_t, t)$ yields

$$dH = \frac{\partial H}{\partial t}\,dt + \frac{\partial H}{\partial x}\,dX_t + \frac{\partial H}{\partial w}\,dw_t + \frac{1}{2}\frac{\partial^2 H}{\partial x^2}\,(dX_t)^2 + \frac{1}{2}\frac{\partial^2 H}{\partial w^2}\,(dw_t)^2,$$

because $dX_t\,dw_t = 0$ (the Brownian motions W_t and B_t^2 are independent).

Compute the quadratic–variation terms

$$(dX_t)^2 = \frac{1}{4}(1-\rho^2)w^2\,dt, \qquad (dw_t)^2 = \sigma^2\,dt,$$

and substitute dX_t, dw_t from the SDEs above. The drift component of dH is

$$\frac{\partial H}{\partial t} + \frac{1}{4}\alpha w^2 \frac{\partial H}{\partial x} + \phi(w)\frac{\partial H}{\partial w} + \frac{1}{8}(1-\rho^2)w^2\frac{\partial^2 H}{\partial x^2} + \frac{1}{2}\sigma^2\frac{\partial^2 H}{\partial w^2}.$$

Under the risk–neutral probability the discounted process $M_t = e^{-rt}H(X_t, w_t, t)$ must be a martingale

Because e^{-rt} is of finite variation, Itô's product yields the following Ito stochastic process representation

$$dM_t = e^{-rt}\,dH_t + H_t\,d(e^{-rt}) = e^{-rt}\Big[dH_t - rH_t\,dt\Big].$$

Now write $dH_t = A_t\,dt + B_t\,d\beta_t$, where A_t is the drift term produced by Itô's lemma and $B_t\,d\beta_t$ collects all Brownian increments. Hence

$$dM_t = e^{-rt}\big[A_t - rH_t\big]dt \;+\; e^{-rt}B_t\,d\beta_t = a_t\,dt + b_t\,d\beta_t,$$

with $a_t := e^{-rt}\big[A_t - rH_t\big]$ and $b_t := e^{-rt}B_t$.

Under the equivalent martingale measure $\mathbb{Q}$ every traded asset discounted by the bank account must be a martingale (Harrison & Kreps, 1979; Harrison & Pliska, 1981) . Therefore

$$\mathbb{E}^{\mathbb{Q}}\big[dM_t \,\big|\, \mathcal{F}_t\big] = a_t\,dt = 0, \quad \text{for all } t \in [0,T].$$

Because dt is nonnegative and greater than 0, we must have $a_t \equiv 0$ almost surely; thus

$$A_t - rH_t = 0 \quad\Longrightarrow\quad A_t = rH_t.$$

Vanishing drift implies

$$\mathbb{E}^{\mathbb{Q}}\big[M_s \,\big|\, \mathcal{F}_t\big] = M_t, \qquad 0 \le t \le s \le T,$$

which is exactly the definition of a martingale adapted to the filtration $\{\mathcal{F}_t\}$. Plugging in rH instead of dH we get following PDE:

$$\frac{\partial H}{\partial t} + \frac{1}{8}(1-\rho^2)w^2\frac{\partial^2 H}{\partial x^2} + \frac{1}{4}\alpha w^2\frac{\partial H}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 H}{\partial w^2} + \phi(w)\frac{\partial H}{\partial w} - rH = 0. \quad (12)$$

Now let our underlying asset follow the regime-switching model discussed in 11. Since the states are discrete and let initial state be n so the option price for underlying stock that follows the can be denoted as $\tilde{H}_n(x,t)$ and inserting $a(P_n), b(P_n)$ to the original discretized equation in 6 we get the following expression Let us formally define payoff of an option that follows Heston dynamics under risk-neutral probability. We fix a regime $n \in \{1,\ldots,N\}$ and define with some payoff $\Phi\big(X_T, w_T\big)$ at log stock price $X_t = x$ and regime $P_t = n$

$$H_n(x,t) \;=\; \mathbb{E}^{\mathbb{Q}}\big[e^{-r\,(T-t)}\,\Phi\big(X_T, w_T\big) \;\big|\; X_t = x,\; P_t = n\big].$$

Over a short interval Δt we can approximate $e^{-r\Delta t} = 1 - r\Delta t$

$$H_n(x,t) = \mathbb{E}^{\mathbb{Q}}\big[(1 - r\Delta t)\,H_{P_{t+\Delta t}}(X_{t+\Delta t},\, t+\Delta t) \mid X_t = x,\; P_t = n\big].$$

By defining property the continious-time Markov chain generator $Q = (q_{n,k})$ for very small Δt,

$$\mathbb{P}(P_{t+\Delta t} = j | P_t = i) = [I + Q\Delta t + o(\Delta t)]_{i,j}$$

and if $i \neq j$ we get:

$$\mathbb{P}(P_{t+\Delta t} = j | P_t = i) = q_{i.j}\Delta t + o(\Delta t)$$

which gives us following formulations of probabilites of switching the regime

$$\Pr\{P_{t+\Delta t} = n \mid P_t = n\} = 1+q_{n,n}\,\Delta t, \quad \Pr\{P_{t+\Delta t} = k \neq n \mid P_t = n\} = q_{n,k}\,\Delta t.$$

meaning that probability of staying in the same state is $1+q_{n,n}\Delta t$ and switching to the states is $1+q_{n,k}\Delta t$
Under $P_t = n$,

$$dX_s = a_n\,ds + b_n\,dW_s, \quad a_n = \tfrac{1}{4}\alpha\,w_n^2, \quad b_n = \tfrac{1}{2}\sqrt{1-\rho^2}\,w_n,$$

so over Δt one has

$$X_{t+\Delta t} - X_t = a_n\,\Delta t + b_n\,\Delta W,$$

$$\mathbb{E}[X_{t+\Delta t} - X_t] = \mathbb{E}[\int_t^{t+\Delta t} a_n ds] + \mathbb{E}[\int_t^{t+\Delta t} b_n dW_s] = a_n\Delta t$$

Since the Expectation of Ito Integral is 0. So the expected increment of X_t is $a_n\Delta t$ To calculate the variance increment, we note that only the stochastic integral contributes to the Variance. By Ito Isometry

$$Var[X_{t+\Delta t} - X_t] = Var[\int_t^{t+\Delta t} b_n dW_s] = b_n^2\Delta t$$

Therefore the Variance increment is $b_n^2\Delta t$

For simplicity take $X_t = x$. Next we have to Taylor Expand $H_k(X_{t+\Delta t}, t+\Delta t)$ around (x,t) For each regime k,

$$\begin{aligned} H_k\big(X_{t+\Delta t}, t+\Delta t\big) &= H_k(x,t) + H_{k,t}\,\Delta t + H_{k,x}\,(X_{t+\Delta t}-x) + \tfrac{1}{2}\,H_{k,xx}\,(X_{t+\Delta t}-x)^2 \\ &\quad + o(\Delta t). \end{aligned}$$

Taking expectations from both sides and using the results above regarding our increments we get the following results. Note that $\Delta W := W_{t+\Delta t} - W_t$ is distributed normallty with mean 0 and variance Δt so this term vanishes

$$\mathbb{E}\big[H_k(X_{t+\Delta t}, t+\Delta t)\big] = H_k + \big(H_{k,t} + a_n H_{k,x} + \tfrac{1}{2}b_n^2 H_{k,xx}\big)\,\Delta t + o(\Delta t).$$

Assembling the every result into one final formula, we don't forget about the probabilites of switching to other regimes and option values at those regimes. Under the risk neutral probability

$$\begin{aligned} H_n(x,t) &= \mathbb{E}^{\mathbb{Q}}\big[(1-r\Delta t)\,H_{P_{t+\Delta t}}(X_{t+\Delta t},\, t+\Delta t) \mid X_t = x,\ P_t = n\big] \\ &= (1-r\,\Delta t)\Big\{(1+q_{n,n}\Delta t)\big[H_n + A_n\,\Delta t\big] + \sum_{k\neq n} q_{n,k}\,\Delta t\ H_k\Big\} + o(\Delta t), \end{aligned}$$

where $A_n := H_{n,t} + a_n H_{n,x} + \frac{1}{2}b_n^2 H_{n,xx}$.

By rearranging terms we get

$$H_n(x,t) = H_n(x,t) + \big[A_n + q_{n,n}H_n + \sum_{k\neq n} q_{n,k}\,\Delta t\ H_k - rH_n\big]\Delta t + o(\Delta t)$$

Dividing every term by Δt and taking its limit to 0:

$$0 = H_{n,t} + a_n\, H_{n,x} + \tfrac{1}{2} b_n^2\, H_{n,xx} + \sum_{k=1}^{N} q_{n,k}\, H_k - r\, H_n,$$

Thus by substituting values of a_n and b_n and representing everything in terms of partial we get our Final PDE:

$$\frac{\partial \tilde{H}_n}{\partial t} + \frac{1}{8}\left(1-\rho^2\right) w^2 \frac{\partial^2 \tilde{H}_n}{\partial x^2} + \frac{1}{4}\alpha w^2 \frac{\partial \tilde{H}_n}{\partial x} + \sum_{k=1}^{N} q_{n,k} \tilde{H}_k - r\tilde{H}_n = 0 \tag{13}$$

Equations 12 and 13 are highly (viewing $\tilde{H}(x,t)$ as $H(x, v_n, t)$) similar so if we can go by the Liu's method of comparing coeficients and simple discretization to variable w, we can easily form the Q-matrix of rates of jumping from one state to another. So discretizing the variable w and setting the unchanging grid with the range $[0, \infty]$ with Δw stands for the space increment. Then let us denote $H_n = H(x, \Delta w, t)$ so we could further approximate our representations of the option price by finite difference method

Applying the central difference scheme for the variance grid:

$$\frac{\partial^2 H}{\partial w^2} \approx \frac{H_{n+1} - 2H_n + H_{n-1}}{(\Delta w)^2},$$
$$\frac{\partial H}{\partial w} \approx \frac{H_{n+1} - H_n}{2\Delta w}.$$

Applying these approximations to the 12 will give us the following equations for the partials w.r.t. w:

$$\frac{1}{2}\sigma^2 \frac{\partial^2 H}{\partial w^2} + \phi(w)\frac{\partial H}{\partial w} = \pi_{n,n+1} H_{n+1} + \pi_{n,n} H_n + \pi_{n,n-1} H_{n-1} \tag{14}$$

$$\pi_{n,n+1} = \frac{\sigma^2}{2(\Delta w)^2} - \frac{4k\theta - \sigma^2}{4(n\Delta w)^2} - \frac{n\kappa}{4},$$
$$\pi_{n,n} = -\frac{\sigma^2}{(\Delta w)^2}$$
$$\pi_{n,n-1} = \frac{\sigma^2}{2(\Delta w)^2} - \frac{4k\theta - \sigma^2}{4(n\Delta w)^2} + \frac{n\kappa}{4}.$$

Thus 14 can be written as

$$\frac{1}{2}\sigma^2 \frac{\partial^2 H}{\partial w^2} + \phi(w)\frac{\partial H}{\partial w} = \sum_{k=1}^{N} \pi_{n,k} H_{n,k} \tag{15}$$

where $\pi_{n,k} = 0$ if $k \neq n-1, n, n+1$. Thus we can see that 12 and 13 are identical and $\tilde{H}_n = H_n$ as $n \to \infty$ The entries for the Q-matrix is almost determined, except the signs of the $\pi_{n,n-1}, \pi_{n,n+1}$ are not exact, i.e they must be non-negative. Zeng [13] proposes the following entries for the Generator matrix with the cases considered in Liu's paper:

Constructing the Q matrix

Case 1

$$\beta_n^+ = \pi_{n,n+1}, \quad \beta_n = -(\pi_{n,n+1} + \pi_{n,n-1}), \quad \beta_n^- = \pi_{n,n-1}$$

Case 2

$$\beta_n^+ = \frac{1}{2}(\pi_{n,n+1} + \pi_{n,n-1}), \quad \beta_n = -2\pi_{n,n-1}, \quad \beta_n^- = -\frac{1}{2}\pi_{n,n+1} + \frac{3}{2}\pi_{n,n-1}$$

Case 3

$$\beta_n^+ = \frac{3}{2}\pi_{n,n+1} - \frac{1}{2}\pi_{n,n-1}, \quad \beta_n = -2\pi_{n,n+1}, \quad \beta_n^- = \frac{1}{2}(\pi_{n,n-1} + \pi_{n,n+1})$$

Extrapolation approach is adopted at the lower and upper bound of regimes with $2H = H_{n+1} + H_{n-1}$ At the lower boundary, the transition rates are:

$$\beta_{n_1}^+ = \pi_{n_1,n_1+1} - \pi_{n_1,n_1-1} = \frac{4\kappa\theta - \sigma^2}{2n_1(\Delta w)^2} - \frac{kn_1}{2} \tag{16}$$

$$\beta_{n_1}^- = \pi_{n_1,n_1} + 2\pi_{n_1,n_1-1} = -(\frac{4\kappa\theta - \sigma^2}{2n_1(\Delta w)^2} - \frac{kn_1}{2}) \tag{17}$$

Similarly at the upper bound

$$\beta_N^+ = \pi_{N,N-1} - \pi_{N,N+1} = \frac{4\kappa\theta - \sigma^2}{2N(\Delta w)^2} - \frac{kN}{2} \tag{18}$$

$$\beta_N = \pi_{N,N} + 2\pi_{N,N-1} = -(\frac{4\kappa\theta - \sigma^2}{2N(\Delta w)^2} - \frac{kN}{2}) \tag{19}$$

We have to choosie a uniform step size Δw. With Δw fixed, we look for an integer $\tilde{n}$ such that

$$f(\tilde{n}) \;=\; 0.$$

Then we pick two numbers for the regimes n_l and N satisfying

$$n_l \;\leq\; \tilde{n} \;\leq\; N.$$

After, the domain $[0, \infty)$ is reduced to the interval $\big[w_{\min}, w_{\max}\big]$, where

$$w_{\min} \;=\; n_l\,\Delta w, \quad w_{\max} \;=\; N\,\Delta w.$$

The grid points $\{w_n\}$ for $n = n_l, \ldots, N$, are obtained giving a total of $N - n_l + 1$ points. Therefore, the approximate Markov chain generator matrix Q can be written as follows:

$$Q = (q_{m,n}) = \begin{pmatrix} \beta^-_{n_l} & \beta^+_{n_l} & 0 & 0 & \cdots & 0 \\ \beta^-_{n_l+1} & \beta_{n_l+1} & \beta^+_{n_l+1} & 0 & \cdots & 0 \\ 0 & \beta^-_{n_l+2} & \beta_{n_l+2} & \beta^+_{n_l+2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \beta^+_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & \beta^+_N \end{pmatrix}.$$

5 Trinomial Tree Approach coupled with Regime-switching approach on Solving the Heston Volatility Model

In a trinomial tree approach for the option pricing model, where the volatility is assumed to be constant, stock can move either up, down or stay the same by a some ratio. Those ratios are calculated and are ass follows: $e^{\sigma\sqrt{\Delta t}}, e^{-\sigma\sqrt{\Delta t}}$ and 1. So in this case Zeng [13] assumes that for the state n, the ratios are $e^{\lambda_n\sigma_n\sqrt{\Delta t}}, e^{-\lambda_n\sigma_n\sqrt{\Delta t}}$ and 1 where $\lambda_n > 1, \sigma_n$ are for each state n, so it is assumed that we consider each regimes one-by-one hence fixing state of Markov Chain. Now we are left with the derivations of the jump probabilities of the stock movement within each state n Let $\phi^u_n, \phi^d_n, \phi^m_n$ be the jump probabilities in the trinomial tree each being up, down and the same respectively. For each tree to have the constant size of a step, $\{\lambda_k\hat{\sigma_k}\}_{k=1}^{k=n} = \tilde{\sigma}$ where Yuen and Yang [14] suggest taking the following: $\tilde{\sigma} = \max_{j=1,2...k}\{\sigma_j\} + (\sqrt{1.5} - 1)\overline{\sigma}$. As in basic probability theory, we now left with taking the first and second moments of the random variable S_t where S_t is stock going either up, down, or staying the same, which match with our volatility and drift terms of the stochastic process

where $\lambda_n > 0$. Match drift μ_n and variance σ_n^2 by solving

$$\begin{cases} \phi^u_n\,(\lambda_n\sigma_n\sqrt{\Delta t}) + \phi^d_n\,(-\lambda_n\sigma_n\sqrt{\Delta t}\,) + \phi^m_n(\Delta t) = \ \mu_n\,\Delta t, \\ \phi^u_n\,(\lambda_n^2\sigma_n^2\,\Delta t) + \phi^d_n\,(\lambda_n^2\sigma_n^2\,\Delta t) + \phi^m_n\,(\Delta t^2) = \ \hat{\sigma_n^2}\,\Delta t, \\ \phi^u_n + \phi^m_n + \phi^d_n \ = \ 1. \end{cases}$$

A suitable solution is

$$\phi^m_n \ = \ 1 \ - \ \frac{1}{\sigma_n^2}, \quad \phi^d_n \ = \ \tfrac{1}{2}\left(\frac{1}{\lambda_n^2} - \frac{\mu_n\sqrt{\Delta t}}{\lambda_n\,\hat{\sigma}_n}\right).$$

Now, it is left to show the connection for all trees in the regime-switching economy, i.e. set up the regime - switching characteristic. As for Zeng [13]reasoning, at each $t = t_i$, the price of a stock can either stay in the same regime (state) or go up and down (u, d) at $t = t_{i+1}$, or it can it can switch regimes (states) but without changing its value (m) at $t = t_i$. So this tree encompasses two changes within one change in a time step. Those are changes in the state (regime) , which is interpreted as a change in a market condition, and change in the price within one state (regime). So there only K+2 nodal values required to compute at each time step against the proposed method of Yuen and Yang [14]. So if we take a node (S_m, t_i, n) it can jump to $(S_{m+1}, t_{i+1}, n), (S_{m-1}, t_{i+1}, n)$ or to $(S_m, t_{i+1}, \hat{n})$ where $\hat{n}$ encompasses all other regimes (states) in this economy Thus we now can define the recursive formula for valuing option price at each node. As discussed above, in a one time step, stock price can either go up or down which is not changed by the switching characteristic in this economy. However, the middle node has the only a non-zero probability of staying in the same regime (state) and switching out to other regimes. Hence, Zeng [13] defines the following recursive formula, which is similar to taking the expectation given the well-defined probabilities, for valuing an option at a time step $t = t_i$:

$$\begin{aligned} U^i_{m,n} = e^{-r\,\Delta t} \Big[\phi^u_n \, U^{i+1}_{m+1,n} \;+\; \big(\phi^m_n + q_{n,n}\,\Delta t\big)\, U^{i+1}_{m,n} \\ +\; \phi^d_n \, U^{i+1}_{m-1,n} \;+\; \sum_{k=1}^{N} q_{n,k}\,\Delta t \, U^{i+1}_{m,k} \Big]. \end{aligned} \tag{20}$$

The transition probability is approximated by $q_{n,k}\Delta t \quad (n \neq k)$ and the probability of staying in the same regime is given by $q_{n,n}\Delta t$ and since in the Q-matrix only the transition probability that takes the state to the same state is given as $q_{n,n} = -\sum_{k \neq k}^{N} q_{n,k}$, the sum of coefficients equals 1.

5.1 New simple trinomial tree approach

Now, we have to apply trinomial tree 20 in the regime-switching economy we defined earlier 11. Amount of regimes in this case becomes $N \to N - n_l + 1$ and $\mu_n \to a_n = a(P_t = n)$, $\hat{\sigma}_n \to b_n = b(P_t = n)$ We should note that X_t follows Brownian motion, whereas S_t follows Geometric Brownian motion as was discussed earlier. We choose $\tilde{b}$ with the same way as chose $\tilde{\sigma}$. $\tilde{b} = \max_{j=1,2\ldots k}\{b_j\} + (\sqrt{1.5} - 1)\bar{b}$, where $\bar{b} = \frac{1}{N - n_l + 1} \sum_{k=n_l}^{N} b_k$. Choose the step sizes as $\tilde{b}\sqrt{\Delta t}, 0, -\tilde{b}\sqrt{\Delta t}$ All the rest follows the same procedure, except that our Q-matrix defined in Section 2 is tridiagonal meaning that the current state n can only jump to either $n - 1$ or $n + 1$, simplifying our original computation of the Option price at the state n

$$\begin{aligned} U^i_{m,n} = e^{-r\,\Delta t} \Big[\phi^u_n \, U^{i+1}_{m+1,n} \;+\; (\phi^m_n + q_{n,n}\,\Delta t)\, U^{i+1}_{m,n} \\ +\; \phi^d_n \, U^{i+1}_{m-1,n} \;+\; q_{n,n-1}\,\Delta t \, U^{i+1}_{m,n-1} \;+\; q_{n,n+1}\,\Delta t \, U^{i+1}_{m,n+1} \Big]. \end{aligned} \tag{21}$$

This construction can be understood straighforwardly. The stock price can either go up and down, but stays in the same regime (the volatility is constant within this regime n). Due to our construction of the Markov chain generator,

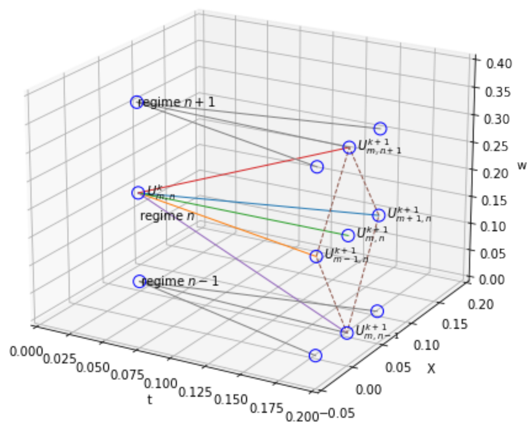

Figure 1: Illustration of a tree with 5 branches starting in a regime n.

if the stock decides to stay the same, it either goes into the state $n-1$ or n as shown in Fig. 1. Let L_k be the # of distinct X-values at each time step k. It can either go up by b grids, go down by b grids or stay at the same level. So $L_0 = 1$ and $L_{k+1} \leq L_k + 2b$ which means $L_k \leq 1 + 2bk$. In a meantime those L_k can occur at each m volatility levels so $Z_k = mL_k \leq m(1+2bk) = m + 2bkm$, which is linear in k. This yields us $Z_N \leq (1 + 2bN) = \mathcal{O}(N)$. This shows that our tree will grow linear, at most $\mathcal{O}(N)$

6 Applying theory onto original coordinate plane

In the approach of the Zeng [13] on construction the simple tree, we got variables X and w, which are purely mathematical and serves no purpose when interpreting them financially. On the other hand, the power of the tree approach is that we can interpret it looking at different branches and using the rebalancing argument at each time step and replicating the option with the use of risk-free bonds and hedging away the risk with the delta shares. Therefore, the main argument of this section is to build the tree in the original setting with more real defined variables, using the price of a stock S_t and the variance v_t that will enhance economic intuition behind this mathematical approach [13].

Rearranging the Beliaeva and Nawalkha [4] transformation on decoupling the two processes, as the jump size was determined at equations 7 and 8, we get the following formulations:

$$\begin{cases} \log \frac{S_t}{S_0} = X_t + \frac{\rho}{\sigma}(v_t - v_0) + \left(r - \frac{\rho\kappa\theta}{\sigma} \right) t, \\ v_t = \frac{1}{4} {w_t}^2. \end{cases} \tag{22}$$

Now, if we consider the stochastic process over a time increment Δt around t

$$\begin{cases} \log \frac{S_{t+\Delta t}}{S_t} = (X_{t+\Delta t} - X_t) + \frac{\rho}{\sigma}\big(v_{t+\Delta t} - v_t\big) + \Big(r - \frac{\rho\kappa\theta}{\sigma}\Big)\Delta t \\ v_{t+\Delta t} - v_t = \frac{1}{4}\big(\, w_{t+\Delta t} + w_t\big)\ \big(\, w_{t+\Delta t} - w_t\big) \end{cases} \tag{23}$$

Intuitively, the X_t and w_t can take three future values - up, middle, down - except at the lower and upper bounds of these variables. Let m, n represent the locations of these two variables at time t, so that we could indicate $X_t = X_m$ and $w_t = w_n$. This allows us to express variables X_t and w_t at time $t + \Delta t$ as the locations in terms of m, n

$$X_{t+\Delta t} = \begin{cases} X_{m+1} & \text{u}, \\ X_m & \text{m}, \\ X_{m-1} & \text{d}, \end{cases} \quad w_{t+\Delta t} = \begin{cases} w_{n+1} & \text{u}, \\ w_n & \text{m}, \\ w_{n-1} & \text{d}. \end{cases}$$

The increments ΔX and Δw are constant which keeps the the structure of the tree simple. The increments of price of a stock S and variance v, however, are simple no more because of $w_{t+\Delta t} + w_t$ at the second part of the equation 23

Rearranging second part of 23 and noting the fact v_t is monotonically increasing, we can define the up, middle and down states of v_t as we did earlier with X and w variables. Using the following expression $w_n = w_{n+1} - \Delta w = w_{n-1} + \Delta w$ we can formally define the states of variable v

$$v_{t+\Delta t} = \begin{cases} v_n + \frac{1}{4}\big(2w_n + \Delta w\big)\,\Delta w, & \text{u}, \\ v_n, & \text{m}, \\ v_n + \frac{1}{4}\big(2w_n - \Delta w\big)\,\Delta w, & \text{d}. \end{cases} \tag{24}$$

Increments of v_t are non-constant, as discussed earlier. Also it should be noticed that the jump size to the up state is greater than the jump size to the down state. To see this, look at the second terms of the up and down states, $\frac{1}{4}(2w_n + \Delta w)\Delta w > \frac{1}{4}(2w_n - \Delta w)\Delta w$

Similarly it is possible to define the up, middle and down states for the stock price at time t, S_t, in accord with the simultaneous movements of the variables X_t and v_t at time $t + \Delta t$. So at time $t + \Delta t$, we define the S_t as follows:

$$S_{t+\Delta t} = \begin{cases} S_n \exp\Big(\Delta X \ + \ \frac{\rho}{\sigma}\,\Delta v \ + \ \Big(r \ - \ \frac{\rho\,\kappa\,\theta}{\sigma}\Big)\Delta t\Big), & \text{up}, \\ S_n \exp\Big(\Big(r \ - \ \frac{\rho\,\kappa\,\theta}{\sigma}\Big)\Delta t\Big), & \text{middle}, \\ S_n \exp\Big(-\Delta X \ - \ \frac{\rho}{\sigma}\,\Delta v \ + \ \Big(r \ - \ \frac{\rho\,\kappa\,\theta}{\sigma}\Big)\Delta t\Big), & \text{down}. \end{cases} \tag{25}$$

7 Results

7.1 Pricing The European Call Option

Zeng [13] gives following two conditions to ensure stability when evaluating the price of an option whose underlying asset follows Heston's stochastic-volatility model:

$$\Delta t < \frac{\Delta w^2}{2\sigma^2} \quad w_{\max} < \frac{\sqrt{2}\,\sigma}{\sqrt{1-\rho^2}}$$

These two conditions are necessary to keep a tree at Fig. 1 simple and stable. Also he states that trinomial tree approach in 22 is equivalent to 12 under $\Delta x = \lambda_n b_n \sqrt{\Delta t}$ Now we can finally define the payoff of at some stock regime m, volatility regime n and at a time $t = t_i$ using 22.

Define the grid at maturity $t_N = T$:

$$\begin{aligned} &X_m = m\,\Delta x, \quad w_n = n\,\Delta w, \quad m = 0,\dots,M,\ n = 0,\dots,N, \\ &v_n = \tfrac{1}{4}\,w_n^2, \quad S_T^{(m,n)} = S_0 \exp\Big(X_m + \tfrac{\rho}{\sigma}\big(v_n - v_0\big) + \big(r - \tfrac{\rho\kappa\theta}{\sigma}\big)\,T\Big). \end{aligned}$$

Set the terminal payoff at each node (m,n) with $i = N$, which makes $N\Delta t = T$:

$$\begin{aligned} U_{m,n}^{i=N} &= \max\big(E - S_T,\,0\big) \\ &= \max\Big\{E \;-\; S_0\,\exp\Big(X_m + \frac{\rho}{\sigma}\,(v_n - v_0) + \big(r - \tfrac{\rho\,\kappa\,\theta}{\sigma}\big)i\Delta t\Big),\ 0 \\ &= \max\Big\{E \;-\; S_0\,\exp\Big(X_m + \frac{\rho}{\sigma}\,(v_n - v_0) + \big(r - \tfrac{\rho\,\kappa\,\theta}{\sigma}\big)T\Big),\ 0\Big\}. \end{aligned}$$

Then we do the backward induction for $i = N-1, N-2, \dots, 0$ using the five–point recursion we defined earlier:

$$\begin{aligned} U_{m,n}^{i} = e^{-r\,\Delta t}\,\Big[&\phi_n^u\,U_{m+1,n}^{i+1} \;+\; (\phi_n^m + q_{n,n}\,\Delta t)\,U_{m,n}^{i+1} \\ &+\; \phi_n^d\,U_{m-1,n}^{i+1} \;+\; q_{n,n-1}\,\Delta t\,U_{m,n-1}^{i+1} \;+\; q_{n,n+1}\,\Delta t\,U_{m,n+1}^{i+1}\Big] \end{aligned}$$

With a numerical solution realized on python at Listing 1, we ready to replicate Zeng's [13] results and compare the analytical results of Heston model and the results of the model discussed in this chapter. Choose

$$r = 0.05, \quad \rho = -0.1, \quad \kappa = 3.0, \quad \theta = 0.04, \quad \sigma = 0.1, \quad E = 100.$$

as model parameters. Now we can run the code for maturities $T = 0.25$ and $T = 0.5$ with initial stock prices $S_0 = 90, 100, 110$ and with two starting variances $v_0 = 0.04, 0.09$. Choose upper and lower bound regimes $n_l = 15, N = 40$, so we took 26 regimes into consideration. Next we should take Δt as low as possible so that the tree is stable. After several experiments, $\Delta t = 10^{-4}$ and

S_0	v_0	T	Benchmark	Tree Price	Error (%)
90	0.04	0.25	0.8852	0.8848	-0.05
100	0.04	0.25	4.6105	4.6094	-0.02
110	0.04	0.25	12.0006	11.9994	-0.01
90	0.09	0.25	1.9023	1.9018	-0.03
100	0.09	0.25	6.0703	6.0682	-0.03
110	0.09	0.25	13.0087	13.0044	-0.03
			$T = 0.50$		
90	0.04	0.50	2.3272	2.3472	0.86
100	0.04	0.50	6.8817	6.9242	0.62
110	0.04	0.50	14.0910	14.1547	0.45
90	0.09	0.50	3.6447	3.6689	0.66
100	0.09	0.50	8.4366	8.4796	0.51
110	0.09	0.50	15.3337	15.3951	0.40

Table 1: European call option prices and percentage-error of the tree approximation

space increment is $\Delta w = 0.02$. Choice of space increment gives us the following variance range:

$$w_{min} = 15 \times 0.02 = 0.30 \quad w_{max} = 40 \times 0.02 = 0.8$$

$$v_{min} = \frac{0.3^2}{4} = 0.0225 \quad v_{max} = \frac{0.8^2}{4} = 0.16$$

We get the following results at Table 1, where benchmark is an analytical solution for Heston model. As we can see the margin of error is at most 1% given the simplicity of Tree structure. Unlike Liu's approach where the Option price at time $t = t_i$ takes into account all the states, in this approach we only take the preceding and succeding states. This gives us much more time efficient computations, but the degree of which it is time efficient will not be discussed. To increase the occuracy, one can choose $\Delta t, \Delta w$ even smaller and increase the number of regimes.

```python
import numpy as np
import math
def heston_tree_euro_call(S0,K,T,r,v0,
                          kappa,theta,sigma_vol,rho,
                          dt,dw,w_lower,w_upper):
    import math
    nsteps=int(round(T/dt))
    w_indices=range(w_lower,w_upper+1)
    w_vals=np.array([j*dw for j in w_indices])
    Nw=len(w_vals)
    alpha=(rho*kappa/sigma_vol)-0.5
    def a_func(w):
        return 0.25*alpha*(w**2)
    def b_func(w):
        return 0.5*np.sqrt(1-rho**2)*w
```

```
    b_list=[b_func(w) for w in w_vals]
    b_max=max(b_list) if b_list else 0
    margin=(math.sqrt(1.5)-1)*b_max
    b_tilde=b_max + margin
    dx =b_tilde * math.sqrt(dt) if b_tilde>1e-15 else 1e-5
    x_min =-4*b_tilde*math.sqrt(T)
    x_max = 4*b_tilde*math.sqrt(T)
    Nx= int(math.ceil((x_max - x_min)/dx)) + 1
    x_vals =np.array([x_min + i*dx for i in range(Nx)])
    U =np.zeros((Nx,Nw), dtype=float)
    def S_func(x, w, t):
        v =0.25 *w**2
        return S0*math.exp(x +(rho/sigma_vol)*(v -v0) + (r - (rho*
    kappa*theta)/sigma_vol)*t)
    for ix in range(Nx):
        for iw in range(Nw):

            S_T =S_func(x_vals[ix], w_vals[iw], T)
            U[ix, iw] =max(S_T - K, 0.0)
    p_up =np.zeros(Nw)
    p_mid= np.zeros(Nw)
    p_dn =np.zeros(Nw)
    for iw in range(Nw):
        w_ =w_vals[iw]
        drift= a_func(w_)
        diff  = b_func(w_)
        m_val     = drift*dt
        var_val= (diff**2)*dt
        denom= dx

        temp =var_val / (dx**2)
        up_=0.5*(temp + m_val/denom)
        dn_= 0.5*(temp - m_val/denom)
        mid_ = 1 - temp
        up_= max(0.0, up_)
        dn_= max(0.0, dn_)
        s_   = up_ + dn_ + mid_
        if s_>1e-15:
            up_/= s_; dn_/= s_; mid_/= s_;
        p_up[iw], p_mid[iw], p_dn[iw] = up_, mid_, dn_
    def phi(w_):
        return (2*kappa*theta - sigma_vol**2/2)/w_ - (kappa/2)*w_
    q_up = np.zeros(Nw)
    q_dn = np.zeros(Nw)
    q_dg = np.zeros(Nw)
    for iw in range(Nw):
        w_= w_vals[iw]
        if iw==0:
            val_up = (sigma_vol**2)/(2*dw**2) + phi(w_)/(dw)
            val_up = max(val_up, 0.0)
            q_up[iw] = val_up
            q_dn[iw]= 0.0
            q_dg[iw] = -(val_up)
        elif iw==Nw-1:
            val_dn =(sigma_vol**2)/(2*dw**2) - phi(w_)/(dw)
            val_dn = max(val_dn, 0.0)
            q_dn[iw]= val_dn
```

```
            q_up[iw] = 0.0
            q_dg[iw] =-(val_dn)
        else:
            val_up =(sigma_vol**2)/(2*dw**2) +phi(w_)/(2*dw)
            val_dn = (sigma_vol**2)/(2*dw**2)- phi(w_)/(2*dw)
            val_up = max(val_up, 0.0)
            val_dn= max(val_dn,0.0)
            q_up[iw] = val_up
            q_dn[iw] = val_dn
            q_dg[iw] =-(val_up+val_dn)
    discount = math.exp(-r*dt)
    for step in range(nsteps):
        U_old = U.copy()
        for ix in range(Nx):
            for iw in range(Nw):
                ix_up   = min(ix+1, Nx-1)
                ix_down = max(ix-1, 0)
                val_x = (p_up[iw]*U_old[ix_up, iw]
                         + p_mid[iw]*U_old[ix, iw]
                         + p_dn[iw]*U_old[ix_down, iw])
                iw_up   = min(iw+1, Nw-1)
                iw_down = max(iw-1, 0)
                val_w = (q_up[iw]*U_old[ix, iw_up]
                         + q_dg[iw]*U_old[ix, iw]
                         + q_dn[iw]*U_old[ix, iw_down]) * dt
                U[ix, iw] = discount*(val_x + val_w)
    w0  = 2.0*math.sqrt(v0)
    iw0 = np.argmin(np.abs(w_vals - w0))
    ix0 = np.argmin(np.abs(x_vals))
    return U[ix0, iw0]

if __name__ == "__main__":
    r= 0.05
    rho= -0.1
    kappa= 3.0
    theta= 0.04
    sigmaH= 0.1
    K = 100.0

    dw= 0.02
    w_lower= 15
    w_upper= 40
    dt = 1e-4
    T_list    = [0.25, 0.5]
    S0_list   = [90, 100, 110]
    v0_list   = [0.04, 0.09]
    print(f" Variables: r={r}, rho=-{rho}, kappa={kappa}, theta={theta}, sigma={sigma}, E={E}")
    print(f" dt={dt}, dw={dw}, w_range=[0.3..0.8], i.e. w_lower={w_lower},w_upper={w_upper}\n")
    print("   S0    v0     T      TreePrice")
    print("  --------------------------------")
    for T in T_list:
        nsteps = int(round(T / dt))
        for s0 in S0_list:
            for vv0 in v0_list:
                tree_price = heston_tree_euro_call(
```

```
                    s0, K, T, r, vv0,
                    kappa, theta, sigmaH, rho,
                    dt, dw, w_lower, w_upper
                )
                print(f"  {s0:3.0f}  {vv0:0.2f}  {T:0.2f}   {
    tree_price:11.4f}")
```

Listing 1: Heston Tree European Call Option

7.2 Pricing the European Call Pricing with Proportional Transaction Costs

Let $C_u^n, C_m^n, C_d^n \geq 0$ be the transaction costs associated with up, middle, and down price moves under regime n, and let $C_{n\to k} \geq 0$ be an additional cost if the Markov chain jumps from regime n to a different regime k. Our cost adjusted Option Pricing model should be in the following representations:

$$\begin{aligned}\widetilde{U}^{\,i}_{m,n} = e^{-r\,\Delta t}\Big[&\phi_n^u\left(U^{i+1}_{m+1,n} - C_u^n\right)\\ &+ \left(\phi_n^m + q_{n,n}\,\Delta t\right)\left(U^{i+1}_{m,n} - C_m^n\right)\\ &+ \phi_n^d\left(U^{i+1}_{m-1,n} - C_d^n\right)\\ &+ \sum_{k\neq n} q_{n,k}\,\Delta t\left(U^{i+1}_{m,k} - C_{n\to k}\right)\Big].\end{aligned} \tag{26}$$

To implement the proportional transaction costs à la Boyle–Vorst [2], we introduce a constant cost-rate λ (e.g. $\lambda = 0.001$ for 0.1% per trade). Whenever we do rebalancing of the portfolio with the delta-hedge between two adjacent tree nodes $(m, n) \to (m', n')$, we pay a cost

$$C_{(m,n)\to(m',n')} \;=\; \lambda\, S_{m,n}\left|\Delta_{m',n'} - \Delta_{m,n}\right|,$$

where

$$\Delta_{m,n} \;=\; \frac{\partial U}{\partial S}\Big(S_{m,n}\Big) \;\approx\; \frac{U_{m+1,n} - U_{m-1,n}}{S_{m+1,n} - S_{m-1,n}} \quad \text{at node } (m,n).$$

In our five-branch tree each node (m, n) has five successors $\{(m\pm1, n), (m, n), (m, n\pm 1)\}$. Denote by $p_{m,n\to m',n'}$ the corresponding jump probability (either $\phi_n^u, \phi_n^m, \phi_n^d$ or $q_{n,n\pm1}\,\Delta t$).

$$p_{m,n\to m',n'} \;=\; \begin{cases} \phi_n^u, & (m', n') = (m+1, n),\\ \phi_n^m, & (m', n') = (m,\, n),\\ \phi_n^d, & (m', n') = (m-1,\, n),\\ q_{n,n+1}\,\Delta t, & (m', n') = (m,\, n+1),\\ q_{n,n-1}\,\Delta t, & (m', n') = (m,\, n-1),\\ 0, & \text{otherwise.}\end{cases}$$

Table 2: European call prices from the Heston tree, with and without transaction costs

S_0	v_0	T	TreePrice (with tc)	TreePrice (without tc)
90	0.04	0.25	0.7436	0.8848
90	0.09	0.25	1.6773	1.9018
100	0.04	0.25	4.2603	4.6094
100	0.09	0.25	5.6747	6.0682
110	0.04	0.25	11.5579	11.9994
110	0.09	0.25	12.5306	13.0044
90	0.04	0.50	1.9943	2.3472
90	0.09	0.50	3.2236	3.6689
100	0.04	0.50	6.2877	6.9242
100	0.09	0.50	7.7974	8.4796
110	0.04	0.50	13.3306	14.1547
110	0.09	0.50	14.5475	15.3951

So the backward recurrence with transaction costs formula reads as follows

$$U^{\,i}_{m,n} = e^{-r\,\Delta t} \sum_{(m',n')\in\mathcal{N}(m,n)} p_{m,n\to m',n'} \Big[U^{\,i+1}_{m',n'} \; - \; \lambda\, S_{m,n} \left|\Delta_{m',n'} - \Delta_{m,n}\right| \Big],$$

where $\mathcal{N}(m,n)$ is the set of five child nodes. In code this is implemented exactly as in Listing 2.

The terminal payoff and all other model parameters remain as before:

$$U^{N}_{m,n} = \max\bigl(E - S_T^{(m,n)}, 0\bigr), \quad S_T^{(m,n)} = S_0 \exp\Bigl(X_m + \tfrac{\rho}{\sigma}(v_n - v_0) + (r - \tfrac{\rho\kappa\theta}{\sigma})T\Bigr).$$

Calculating the cost-adjusted option price with the backward induction from $i = N-1$ down to $i = 0$, we easily obtain the European call price inclusive of proportional trading costs. A numerical solution is given in Listing 2 as a python code, and the resulting prices for the 12 test cases appear in Table 2. As we don't have the analytical benchmark for the heston option pricing model with added transaction costs, we don't have accurate proxy for that. However we still can compare the results with the plain trinomial tree under regime-switching economy without the transaction costs. We see that they results vary, but the still stand in the neighborhood of the tree results without the transaction costs

```
import numpy as np
import math

def heston_tree_euro_call_tc(S0, K, T, r, v0,
                             kappa, theta, sigma_vol, rho,
                             dt, dw, w_lower, w_upper,
                             tc_rate=0.001):
    nsteps = int(round(T / dt))
    w_indices = range(w_lower, w_upper+1)
    w_vals = np.array([j*dw for j in w_indices])
    Nw = len(w_vals)
```

```python

    alpha =(rho * kappa / sigma_vol) - 0.5
    def a_func(w):
        return 0.25 * alpha * (w**2)
    def b_func(w):
        return 0.5 *math.sqrt(1 - rho**2) * w

    b_list =[b_func(w) for w in w_vals]
    b_max= max(b_list) if b_list else 0.0
    margin= (math.sqrt(1.5) - 1.0) * b_max
    b_tilde =b_max + margin
    dx = b_tilde * math.sqrt(dt) if b_tilde > 1e-15 else 1e-5

    x_min =-4 * b_tilde * math.sqrt(T)
    x_max=  4 * b_tilde * math.sqrt(T)
    Nx =int(math.ceil((x_max - x_min) / dx)) + 1
    x_vals = np.array([x_min + i * dx for i in range(Nx)])

    U =np.zeros((Nx, Nw), dtype=float)

    def S_func(x, w, t):
        v = 0.25 * w**2
        return S0 * math.exp(x + (rho/sigma_vol) * (v - v0) + (r -
    (rho*kappa*theta)/sigma_vol) * t)

    for ix in range(Nx):
        for iw in range(Nw):
            S_T = S_func(x_vals[ix], w_vals[iw], T)
            U[ix, iw] = max(S_T - K, 0.0)

    p_up= np.zeros(Nw)
    p_mid =np.zeros(Nw)
    p_dn = np.zeros(Nw)
    for iw in range(Nw):
        w_= w_vals[iw]
        m_val= a_func(w_) * dt
        var_val= (b_func(w_)**2) * dt
        temp= var_val / (dx**2)
        up_= 0.5 * (temp + m_val/dx)
        dn_  = 0.5 * (temp - m_val/dx)
        mid_ = 1.0 - temp

        up_ = max(up_, 0.0)
        dn_= max(dn_, 0.0)
        s_ = up_ + dn_ + mid_
        if s_ > 1e-15:
            up_  /= s_

            dn_ /= s_
            mid_ /= s_
        p_up[iw], p_mid[iw], p_dn[iw] = up_, mid_, dn_

    def phi(w_):
        return (2*kappa*theta - sigma_vol**2/2)/w_ - (kappa/2)*w_

    q_up= np.zeros(Nw)
    q_dn = np.zeros(Nw)
```

```python
    q_dg= np.zeros(Nw)
    for iw in range(Nw):
        w_ = w_vals[iw]
        if iw == 0:
            val_up =(sigma_vol**2) / (2*dw**2) + phi(w_) / dw
            val_up= max(val_up, 0.0)
            q_up[iw]= val_up
            q_dn[iw] = 0.0
            q_dg[iw] = -val_up
        elif iw== Nw - 1:
            val_dn = (sigma_vol**2) / (2*dw**2) - phi(w_) / dw
            val_dn = max(val_dn, 0.0)
            q_dn[iw] = val_dn
            q_up[iw]= 0.0
            q_dg[iw] = -val_dn
        else:
            val_up =(sigma_vol**2) / (2*dw**2) + phi(w_) / (2*dw)
            val_dn = (sigma_vol**2) / (2*dw**2) - phi(w_) / (2*dw)
            val_up= max(val_up, 0.0)
            val_dn = max(val_dn, 0.0)
            q_up[iw] = val_up
            q_dn[iw]= val_dn
            q_dg[iw] = -(val_up + val_dn)

    disc = math.exp(-r * dt)

    def get_delta(ix, iw, Uarr, t):
        if ix ==0:
            dU = Uarr[ix+1, iw] - Uarr[ix, iw]
            dS =S_func(x_vals[ix+1], w_vals[iw], t) - S_func(x_vals[ix], w_vals[iw], t)
        elif ix == Nx - 1:
            dU= Uarr[ix, iw] - Uarr[ix-1, iw]
            dS =S_func(x_vals[ix], w_vals[iw], t) - S_func(x_vals[ix-1], w_vals[iw], t)
        else:
            dU = Uarr[ix+1, iw] - Uarr[ix-1, iw]
            dS = S_func(x_vals[ix+1], w_vals[iw], t) - S_func(x_vals[ix-1], w_vals[iw], t)
        return 0.0 if abs(dS) < 1e-12 else dU / dS

    def rebalance_cost(ixp, iwp, ixc, iwc, Uarr, t):
        delta_parent = get_delta(ixp, iwp, Uarr, t)
        delta_child  = get_delta(ixc, iwc, Uarr, t)
        S_parent = S_func(x_vals[ixp], w_vals[iwp], t)
        return tc_rate * abs(delta_child - delta_parent) * S_parent

    for step in range(nsteps):
        U_old = U.copy()
        t = T - step * dt
        for ix in range(Nx):
            ix_up= min(ix+1, Nx-1)
            ix_down = max(ix-1, 0)
            for iw in range(Nw):
                val_x = (p_up[iw] * U_old[ix_up, iw] +
                         p_mid[iw] * U_old[ix, iw] +
                         p_dn[iw] * U_old[ix_down, iw])
```

```
                cost_x = (p_up[iw] * rebalance_cost(ix, iw, ix_up, iw,  U_old, t) +
                          p_mid[iw] * rebalance_cost(ix, iw, ix, iw,  U_old, t) +
                          p_dn[iw] * rebalance_cost(ix, iw, ix_down, iw,  U_old, t))
                iw_up= min(iw+1, Nw-1)
                iw_down = max(iw-1, 0)
                val_w = (q_up[iw] * U_old[ix, iw_up] +
                         q_dg[iw] * U_old[ix, iw]    +
                         q_dn[iw] * U_old[ix, iw_down]) * dt
                cost_w = (q_up[iw] * rebalance_cost(ix, iw, ix, iw_up,   U_old, t) +
                          q_dg[iw] * rebalance_cost(ix, iw, ix, iw,   U_old, t) +
                          q_dn[iw] * rebalance_cost(ix, iw, ix, iw_down,  U_old, t)) * dt
                U[ix, iw] = disc * ((val_x + val_w) - (cost_x + cost_w))

    w0  = 2.0 * math.sqrt(v0)
    iw0= np.argmin(np.abs(w_vals - w0))
    ix0 =np.argmin(np.abs(x_vals))
    return U[ix0, iw0]

if __name__ == "__main__":
    r= 0.05
    rho= -0.1
    kappa= 3.0
    theta=0.04
    sigmaH= 0.1
    K= 100.0
    dw= 0.02
    w_lower =15
    w_upper= 40
    tc_rate = 0.0005
    T_list= [0.25, 0.5]
    S0_list= [90, 100, 110]
    v0_list =[0.04, 0.09]
    dt = 1e-3

    print(f"  r={r}, rho={rho}, kappa={kappa}, theta={theta}, sigma={sigma}, E={E}")
    print(f"  dt={dt}, dw={dw}, w_range=[15,40]; tc_rate=0.001\n")
    print("   S0    v0     T      TreePrice (with tc)")
    print("  ----------------------------------------")
    for T in T_list:
        for s0 in S0_list:
            for vv0 in v0_list:
                price = heston_tree_euro_call_tc(
                    s0, K, T, r, vv0,
                    kappa, theta, sigmaH, rho,
                    dt, dw, w_lower, w_upper,
                    tc_rate=tc_rate
                )
                print(f"  {s0:3.0f}  {vv0:0.2f}  {T:0.2f}   {price
```

```
:11.4f}")
```

Listing 2: Heston Tree European Call Option with added Transaction costs

8 Conclusion

Starting from an regime switching argument - a core to this capstone - we were able to derive the approximation for a closed form analytical soltuion to the Heston PDE with an illustration, which is more efficient than other multinomial models, because it grows linearly, dependent on the which state the tree is on, . Later we were able to replicate Zeng's [13] results and also add the transaction costs according to Boyle and Vorst' idea [2]. For each of the cases, the numerical results, as well as the python code, is provided. For the future Research on the regime-switching models, I would focus on American Options as the trinomial trees are widely used to price them.

References

[1] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy **81**(3) (1973), 637–654.

[2] P. P. Boyle and T. Vorst, *Option Replication in Discrete Time with Transaction Costs*, The Journal of Finance **47**(1) (1992), 271–293.

[3] J. C. Cox, S. A. Ross, and M. Rubinstein, *Option Pricing: A Simplified Approach*, Journal of Financial Economics **7**(3) (1979), 229–263.

[4] N. A. Beliaeva and S. K. Nawalkha, *A Simple Approach to Pricing American Options Under the Heston Stochastic Volatility Model*, Journal of Derivatives **17**(4) (2010), 25–43.

[5] S. L. Heston, *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, Review of Financial Studies **6**(2) (1993), 327–343.

[6] Hilliard, J. E., and A. Schwartz, *Binomial option pricing under stochastic volatility and correlated state variables*, Journal of Derivatives, Fall (1996). Available at SSRN: https://ssrn.com/abstract=7691

[7] J. M. Harrison and D. M. Kreps, *Martingales and arbitrage in multiperiod securities markets*, Journal of Economic Theory **20**(3) (1979), 381–408.

[8] J. M. Harrison and S. R. Pliska, *Martingales and stochastic integrals in the theory of continuous trading*, Stochastic Processes and their Applications **11**(3) (1981), 215–260.

[9] Z. Janková, *Drawbacks and Limitations of Black–Scholes Model for Options Pricing*, Journal of Financial Studies & Research **2018** (2018), 1–7.

[10] H. E. Leland, *Option Pricing and Replication with Transactions Costs*, Journal of Finance **40**(5) (1985), 1283–1301.

[11] R. H. Liu, *Regime-Switching Recombining Tree for Option Pricing*, International Journal of Theoretical and Applied Finance **13**(3) (2010), 479–499.

[12] D. B. Nelson and K. Ramaswamy, *Simple Binomial Processes as Diffusion Approximations in Financial Models*, Review of Financial Studies **3**(3) (1990), 393–430.

[13] X.-C. Zeng and S.-P. Zhu, *A new simple tree approach for the Heston's stochastic volatility model*, Computers & Mathematics with Applications **78**(6) (2019), 1993–2010.

[14] F. L. Yuen and H. Yang, *Option pricing with regime switching by trinomial tree method*, Journal of Computational and Applied Mathematics **233**(8) (2010), 1821–1833.