Options Pricing under Numerical Heston PDE

AUTHORS:

Xuyu Cai

Yupeng Lu

Dejiang Zhang

Caleb Qi

Chunlu Zhang

Chunxiang Wang

MAY 2022

Abstract

We priced European SPY options under the Heston dynamic using PDE method**.** First, we pick the proper discretization grid for the underlying asset price S_i, volatility v_j and time T_k. Then central finite difference approximations (implicit scheme) are used for each partial derivative in the PDE. Next, boundary conditions of the PDE are specified to solve the PDE. By doing these, calculations are vectorized in matrices to numerically backward induce the option prices.

Introduction

Among the methods of options pricing discussed in class, one of the more mathematically oriented methods involves discretizing the Heston PDE to numerically (implicit or explicit scheme) solve the PDE and price European call options on SPY. This method is generally used for exotic options, as it allows us to know the payoff pattern of the option at points in grided time, the underlying asset and volatility.

Heston Model

The Heston Model is supposed to be an improvement to the Black-Scholes model which had taken some assumptions which were unrealistic. The main assumption of the BS model being that volatility remained constant over the period of the option lifetime is restated in the Heston model. In the BS model, volatility is kept as a value that can not be predicted and follows a random process. This model gives us a closed-form solution that greatly simplified the process and led to greater adoption. On the other hand, the Heston gives us the volatility which follows a stochastic process that allows us to fit the volatility skew in the real market. However, this improvement makes the calculation more complex because the parameters have to be calibrated carefully to provide a decent estimate of the option prices. Further, it is found that the

Heston model suffers when it comes to predicting the option prices for the short-term options as the model fails to capture the high implied. It is also comparatively more complex than the Black Scholes model which deters from using this option.

The Heston PDE

The Heston PDE is derived in Hirsa's textbook roughly with the following steps, which we have included here briefly for the sake of completeness:

1. We start with the Heston model SDE:

$$dS(t) = \mu S dt + \sqrt{v(t)} S dz_1(t) \qquad (1)$$

$$d\sqrt{v(t)} = -\beta\sqrt{v(t)}dt + \delta dz_2(t) \qquad (2)$$

2. We transform the Heston SDEs (volatility process) into an OU process by applying Ito's lemma.
3. An arbitrage argument is applied to convert the OU process SDEs into PDEs.
4. By switching to risk-neutral probabilities, we can eliminate the price of volatility risk and end up with the Heston PDE.

The Heston PDE Formula

$$\frac{1}{2}vS^2\frac{\partial^2 U}{\partial S^2} + \rho\sigma v S\frac{\partial^2 U}{\partial S \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 U}{\partial v^2} + (r-q)S\frac{\partial U}{\partial S} + (\kappa(\theta - v) - \lambda v)\frac{\partial U}{\partial v} - rU - \frac{\partial U}{\partial \tau} = 0 \qquad (3)$$

We need to find the 6 partial derivatives. This is usually done via discretization, but since the functions for S and nu were explicitly stated in Hirsa, we used the scipy derivative function to obtain the derivatives. We obtained the parameters in the PDE from calibrating the Heston model with European SPY options data on 4/1/2022 from Bloomberg.

The Boundary Conditions

$$U(S,v,0) = (S-K)^+ \qquad (4)$$

$$\lim_{S \downarrow 0} \frac{\partial^2 U}{\partial S^2}(S,v,\tau) = 0 \qquad (5)$$

$$\lim_{S \uparrow \infty} \frac{\partial^2 U}{\partial S^2}(S,v,\tau) = 0 \qquad (6)$$

$$(r-q)S\frac{\partial U}{\partial S}(S,0,\tau) + \kappa\theta\frac{\partial U}{\partial v}(S,0,\tau) - rU(S,0,\tau) - \frac{\partial U}{\partial \tau}(S,0,\tau) = 0 \qquad (7)$$

$$\lim_{v \uparrow \infty} U(S,v,\tau) = S \qquad (8)$$

The first condition is simply the option price. At any given time, the price of the option is the max of the underlying price minus the strike and 0. The second and third conditions state that as the underlying price goes to infinity and 0, the gamma for the option goes to 0. This can be explained via the gamma formula or simply that the rate of change, or the rate of change of the option, goes to 0 when the price is extremely low or extremely high. The fourth condition is simply the PDE when volatility is 0. The last boundary condition states that when volatility is infinite, the option becomes the underlying asset via the arbitrage argument.

Methodology

Our project consists of the following steps:

1. Calibration of Heston model: Given stochastic model pricing data for European options from Bloomberg, we calibrated the Heston model to obtain the parameters for the PDE.
2. Creation of the implicit scheme: This is the matrix that we used to obtain the payoff vector of the previous time spot
3. Implementing the boundary conditions: We need to append our price vectors according to our boundary condition assumptions.

4. Obtaining a matrix of price: Multiplying a price vector at each step and adding boundary values.

The details for 2 - 4 are discussed below.

Numerical Scheme

In Hirsa's textbook, a transformation is applied that shrinks the domain of our volatility grid and our strike grid to [0,1]. As a result, we obtain the following new PDEs and new Boundary conditions. That being said, they carry the same meaning as before.

Transformation

$$S(\xi) = K + \alpha sinh(c_1\xi + c_2(1-\xi)) \qquad (9)$$

$$v(\eta) = \beta sinh(d\eta) \qquad (10)$$

$$c_1 = sinh^{-1}(\frac{S_{max} - K}{\alpha}) \qquad (11)$$

$$c_2 = sinh^{-1}(\frac{S_{min} - K}{\alpha}) \qquad (12)$$

$$d = sinh^{-1}(\frac{v_{max}}{\beta}) \qquad (13)$$

Transformed PDE

$$\begin{aligned}\frac{1}{2}v(\eta)\ S^2(\xi)\left(\frac{\partial^2\bar{U}}{\partial\xi^2}\frac{1}{(\frac{\partial S}{\partial\xi})^2} - \frac{\partial\bar{U}}{\partial\xi}\frac{\frac{\partial^2 S}{\partial\xi^2}}{(\frac{\partial S}{\partial\xi})^3}\right) + \rho\sigma v(\eta)S(\xi)\frac{\partial^2\bar{U}}{\partial\xi\partial\eta}\frac{1}{\frac{\partial S}{\partial\xi}\frac{\partial v}{\partial\eta}} + \\ \frac{1}{2}\sigma^2 v(\eta)\left(\frac{\partial^2\bar{U}}{\partial\eta^2}\frac{1}{(\frac{\partial v}{\partial\eta})^2} - \frac{\partial\bar{U}}{\partial\eta}\frac{\frac{\partial^2 v}{\partial\eta^2}}{(\frac{\partial v}{\partial\eta})^3}\right) + (r-q)S(\xi)\frac{\partial\bar{U}}{\partial\xi}\frac{1}{\frac{\partial S}{\partial\xi}} + \\ \kappa(\theta - v(\eta))\frac{\partial\bar{U}}{\partial\eta}\frac{1}{\frac{\partial v}{\partial\eta}} - r\bar{U} - \frac{\partial\bar{U}}{\partial\tau} = 0\end{aligned} \qquad (14)$$

Transformed Boundary Conditions

$$\bar{U}(\xi,\eta,0) = (S(\xi) - K)^+ \qquad (15)$$

$$\frac{1}{(\frac{\partial S}{\partial\xi})^2}\frac{\partial^2\bar{U}}{\partial\xi^2}(0,\eta,\tau) - \frac{\frac{\partial^2 S}{\partial\xi^2}}{(\frac{\partial S}{\partial\xi})^3}\frac{\partial\bar{U}}{\partial\xi}(0,\eta,\tau) = 0 \qquad (16)$$

$$\frac{1}{(\frac{\partial S}{\partial \xi})^2}\frac{\partial^2 \bar{U}}{\partial \xi^2}(1,\eta,\tau) - \frac{\frac{\partial^2 v}{\partial \eta^2}}{(\frac{\partial v}{\partial \eta})^3}\frac{\partial \bar{U}}{\partial \xi}(1,\eta,\tau) = 0 \quad (17)$$

$$(r-q)S(\xi)\frac{1}{\frac{\partial S}{\partial \xi}}\frac{\partial \bar{U}}{\partial \xi}(\xi,0,\tau)+\kappa\theta\frac{1}{\frac{\partial v}{\partial \eta}}\frac{\partial \bar{U}}{\partial \eta}(\xi,0,\tau)-r\bar{U}(\xi,0,\tau)-\frac{\partial \bar{U}}{\partial \tau}(\xi,0,\tau) = 0 \quad (18)$$

$$\bar{U}(\xi,1,\tau) = S(\xi) \quad (19)$$

Grids:

$$\bar{D} = \left\{ \begin{array}{lll} \xi_i = 0 + (i-1)\Delta\xi; & \Delta\xi = \frac{1}{N}; & i = 1,\ldots,N+1 \\ \eta_j = 0 + (j-1)\Delta\eta; & \Delta\eta = \frac{1}{M}; & j = 1,\ldots,M+1 \\ \tau_k = 0 + (k-1)\Delta\tau; & \Delta\tau = \frac{T-0}{L}; & k = 1,\ldots,L+1 \end{array} \right\}$$

Approximations for Partial Derivatives

As previously mentioned, instead of using finite difference approximations for the partial derivatives, we used the derivative function to obtain the derivatives of the scipy.misc package. This is also possible because of the linear to hyperbolic transformation mentioned before. If the grid functions were linear, the derivatives would be constant. Below are the partial derivatives in the discretization approximation form.

$$\frac{\partial^2 \bar{U}}{\partial \xi^2}(\xi,\eta_j,\tau_{k+1}) = \frac{U_{i-1,j}^{k+1} - 2U_{i,j}^{k+1} + U_{i+1,j}^{k+1}}{\Delta\xi^2} + O(\Delta\xi^2) \quad (20)$$

$$\frac{\partial^2 \bar{U}}{\partial \eta^2}(\xi,\eta_j,\tau_{k+1}) = \frac{U_{i,j-1}^{k+1} - 2U_{i,j}^{k+1} + U_{i,j+1}^{k+1}}{\Delta\eta^2} + O(\Delta\eta^2) \quad (21)$$

$$\frac{\partial^2 \bar{U}}{\partial \xi \partial \eta}(\xi,\eta_j,\tau_{k+1}) = \frac{U_{i-1,j-1}^{k+1} - U_{i-1,j+1}^{k+1} - U_{i+1,j-1}^{k+1} + U_{i+1,j+1}^{k+1}}{4\Delta\xi\Delta\eta} + O(\Delta\eta^2) + O(\Delta\xi^2) \quad (22)$$

$$\frac{\partial \bar{U}}{\partial \xi}(\xi, \eta_j, \tau_{k+1}) = \frac{U_{i+1,j}^{k+1} - -U_{i-1,j}^{k+1}}{2\Delta\xi} + O(\Delta\xi^2) \tag{23}$$

$$\frac{\partial \bar{U}}{\partial \eta}(\xi, \eta_j, \tau_{k+1}) = \frac{U_{i,j+1}^{k+1} - -U_{i,j-1}^{k+1}}{2\Delta\eta} + O(\Delta\eta^2) \tag{24}$$

$$\frac{\partial \bar{U}}{\partial \tau}(\xi, \eta_j, \tau_{k+1}) = \frac{U_{i,j}^{k+1} - -U_{i,j}^{k+1}}{\Delta\tau} + O(\Delta\tau) \tag{25}$$

From the formulas for the six partial derivatives, we have that at each given time, for each combination of strike and PDE grid nodes, (i, j), we need to use the following elements of the price matrix U.

$$\begin{matrix} U_{i-1,j+1}^{k+1} & U_{i,j+1}^{k+1} & U_{i+1,j+1}^{k+1} \\ U_{i-1,j}^{k+1} & U_{i,j}^{k+1} & U_{i+1,j}^{k+1} \\ U_{i-1,j-1}^{k+1} & U_{i,j-1}^{k+1} & U_{i+1,j-1}^{k+1} \end{matrix}$$

Difference Equation

After plugging the above partial derivative approximation formulas into the PDE, we then obtain a difference equation from the Heston PDE. We use this difference equation to get from time k+1 to time k. Just like in the lecture. Again, we are going backwards because we are using the implicit method. A lot of algebra manipulation is involved, but long story short, this is what thedifference equation looks like:

$$\begin{aligned} -a_{i,j}U_{i-1,j-1}^{k+1} - b_{i,j}U_{i,j-1}^{k+1} + a_{i,j}U_{i+1,j-1}^{k+1} - c_{i,j}U_{i-1,j}^{k+1} + d_{i,j}U_{i,j}^{k+1} \\ -e_{i,j}U_{i+1,j}^{k+1} + a_{i,j}U_{i-1,j+1}^{k+1} - f_{i,j}U_{i,j+1}^{k+1} - a_{i,j}U_{i+1,j+1}^{k+1} = U_{i,j}^{k} \end{aligned} \tag{26}$$

Where

$$
\begin{aligned}
a_{i,j} &= \frac{\Delta\tau\rho\sigma}{4\Delta\xi\Delta\eta} v(\eta_j) S(\xi_i) \frac{1}{\frac{\partial S}{\partial \xi}(\xi_i) \frac{\partial v}{\partial \eta}(\eta_j)} \\
b_{i,j} &= \frac{\sigma^2\Delta\tau}{2(\Delta\eta)^2} v(\eta_j) \frac{1}{(\frac{\partial v}{\partial \eta}(\eta_i))^2} - \frac{\Delta\tau}{2\Delta\eta}\Big(\kappa(\theta - v(\eta_j)) \frac{1}{\frac{\partial v}{\partial \eta}(\eta_j)} - \frac{1}{2}\sigma^2 v(\eta_j) \frac{\frac{\partial^2 v}{\partial \eta^2}(\eta_j)}{(\frac{\partial v}{\partial \eta}(\eta_j))^3}\Big) \\
c_{i,j} &= \frac{\Delta\tau}{(\Delta\xi)^2} v(\eta_j) S(\xi_i)^2 \frac{1}{(\frac{\partial S}{\partial \xi}(\xi_i))^2} - \frac{\Delta\tau}{2\Delta\xi}\Big((r-q) S(\xi_i) \frac{1}{\frac{\partial S}{\partial \xi}(\xi_i)} - \frac{1}{2} v(\eta_j) S^2(\xi_i) \frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_i)}{(\frac{\partial S}{\partial \xi}(\xi_i))^3}\Big) \\
d_{i,j} &= 1 + r\Delta\tau + \frac{\Delta\tau}{(\Delta\xi)^2} v(\eta_j) S(\xi_i)^2 \frac{1}{(\frac{\partial S}{\partial \xi}(\xi_i))^2} + \frac{\sigma^2\Delta\tau}{(\Delta\eta)^2} v(\eta_j) \frac{1}{(\frac{\partial v}{\partial \eta}(\eta_i))^2} \\
e_{i,j} &= \frac{\Delta\tau}{(\Delta\xi)^2} v(\eta_j) S(\xi_i)^2 \frac{1}{(\frac{\partial S}{\partial \xi}(\xi_i))^2} + \frac{\Delta\tau}{2\Delta\xi}\Big((r-q) S(\xi_i) \frac{1}{\frac{\partial S}{\partial \xi}(\xi_i)} - \frac{1}{2} v(\eta_j) S^2(\xi_i) \frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_i)}{(\frac{\partial S}{\partial \xi}(\xi_i))^3}\Big) \\
f_{i,j} &= \frac{\sigma^2\Delta\tau}{2(\Delta\eta)^2} v(\eta_j) \frac{1}{(\frac{\partial v}{\partial \eta}(\eta_i))^2} + \frac{\Delta\tau}{2\Delta\eta}\Big(\kappa(\theta - v(\eta_j)) \frac{1}{\frac{\partial v}{\partial \eta}(\eta_j)} - \frac{1}{2}\sigma^2 v(\eta_j) \frac{\frac{\partial^2 v}{\partial \eta^2}(\eta_j)}{(\frac{\partial v}{\partial \eta}(\eta_j))^3}\Big)
\end{aligned}
$$

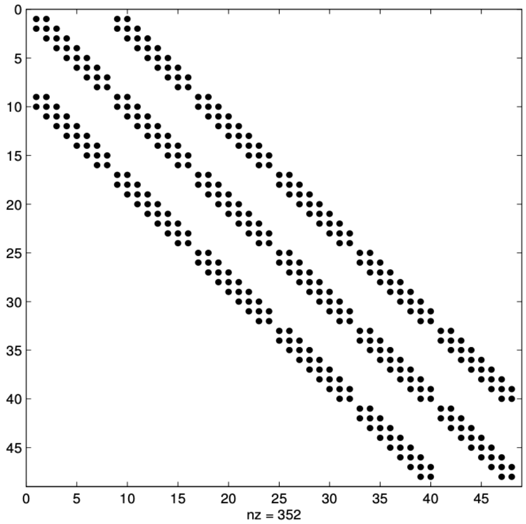
As we can see, the difference equation involves 9 different elements, which will end up being the 9 different diagonals on the matrix A. We can once again draw inspiration from our lecture, where we had a tridiagonal A matrix and the diagonals were l, d, and u. In this step, similar to what we did in lecture, we will end up expression our difference equation in matrix form so that we can iterate forward from k to k + 1 in time by left matrix multiplication of A inverse as such:

$$AU^{k+1} = U^k + r.h.s. \qquad (27)$$

The r.h.s corresponds to the boundary conditions which we will include to our output at each iteration.

Matrix

Given the way the strikes, the PDE, and the boundary conditions are defined so far, instead of having just one strike dimension in our lecture, we will have an extra volatility grid dimension in our price matrix. We can express this either as a 3D matrix or as a tridiagonal block matrix as shown in Hirsa. This is what the matrix and the vector we are multiplying it with looks like:



$$\mathbf{U^{k+1}} = \begin{pmatrix} U_{2,2}^{k+1} \\ U_{3,2}^{k+1} \\ \vdots \\ U_{N,2}^{k+1} \\ U_{2,3}^{k+1} \\ U_{3,3}^{k+1} \\ \vdots \\ U_{N,3}^{k+1} \\ \vdots \\ U_{2,M}^{k+1} \\ U_{3,M}^{k+1} \\ \vdots \\ U_{N,M}^{k+1} \end{pmatrix}$$

On the left is a block tridiagonal matrix. It is of $(M - 1) * (N - 1)$ dimension because a, b, c, d, e, and f vary over xi(strike) and eta(volatility) dimensions, as opposed to in the lecture, where a, l, and u only vary over the strike dimension. Note that on the diagonals, we have blocks of tridiagonal matrices, which themselves make up a tridiagonal matrix. For the atomic matrices on the diagonal, each column corresponds to a strike, but, for the grand matrix, the difference is that each column corresponds to a different strike.

On the right, we have a vector of length $(M - 1) * (N - 1)$. Observe that the first index of the elements corresponds to the indices of the columns of the atomic matrices on the diagonal of A, and the second index of the elements corresponds to the indices of the matrices themselves. As in, 2 being the 2nd matrix, 3 being the 3rd matrix, and so on, as opposed to 2 being the 2nd column, 3 being the 3rd column, and so on. We can see this because when we multiply a vector U by the matrix A, the first index matches with the columns of the atomic matrices, while the second index matches with the columns of the grand matrix (the indices of the atomic matrices.

Also note that there are “holes” in the matrix. First we point out the holes on the upper left corner and lower right corner of every atomic matrix on the diagonal. Then for the grand matrix, we are missing two atomic tridiagonal matrices, one at the top left corner, and one at the bottom right corner. If we imagine the grand matrix as being composed of 9 equal-length diagonals extending off the square and glued to the bottom and the top of the squares, we can see how these holes could be filled. These “holes” correspond precisely to the boundary conditions, which make up the “r.h.s” part of the matrix multiplication scheme. But we are not looking to fill up the matrix A, we will be adding the boundary conditions to the resulting vectors.

Boundary Conditions

There are 5 boundary conditions. The first one is already incorporated into the matrix, since it is the price function of our option. In a nutshell, the first two boundary conditions correspond to the holes at the top left and bottom right of each atomic tridiagonal matrix on the grand tridiagonal matrix. Below are the formulas used in the boundary conditions, expressed in the form of difference equations.

For the first boundary condition corresponding to i = 2, we have

$$\begin{aligned}(-a_{2,j}\underline{l}_2 - b_{2,j})U^{k+1}_{2,j-1} + (-a_{2,j}\bar{l}_2 + a_{2,j})U^{k+1}_{3,j-1} + (-c_{2,j}\underline{l}_2 + d_{2,j})U^{k+1}_{2,j} \\ +(-c_{2,j}\bar{l}_2 - e_{2,j})U^{k+1}_{3,j} + (a_{2,j}\underline{l}_2 - f_{2,j})U^{k+1}_{2,j+1} + (a_{2,j}\bar{l}_2 - a_{2,j})U^{k+1}_{3,j+1} \quad &= \quad U^k_{2,j} \end{aligned} \qquad (28)$$

Where:

$$\underline{l}_2 = \frac{\frac{2}{(\frac{\partial S}{\partial \xi}(\xi_2))^2}}{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_2))^2} + \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_2)}{(\frac{\partial S}{\partial \xi}(\xi_2))^3}} \qquad (29)$$

$$\underline{l}_2 = -\frac{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_2))^2} - \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_2)}{(\frac{\partial S}{\partial \xi}(\xi_2))^3}}{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_2))^2} + \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_2)}{(\frac{\partial S}{\partial \xi}(\xi_2))^3}} \qquad (30)$$

For the boundary condition corresponding to i = N, we have

$$
\begin{aligned}
(-a_{N,j} + a_{N,j}\underline{r}_N)U^{k+1}_{N-1,j-1} + (-b_{N,j} + a_{N,j}\bar{r}_N)U^{k+1}_{N,j-1} \\
+(-c_{N,j} - e_{N,j}\underline{r}_N)U^{k+1}_{N-1,j} + (d_{N,j} - e_{N,j}\bar{r}_N)U^{k+1}_{N,j} \\
+(a_{N,j} - a_{N,j}\underline{r}_N)U^{k+1}_{N-1,j+1} + (-f_{N,j} - a_{N,j}\bar{r}_N)U^{k+1}_{N,j+1} &= U^k_{N,j}
\end{aligned} \tag{31}
$$

Where

$$
\underline{r}_N = -\frac{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_N))^2} + \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_N)}{(\frac{\partial S}{\partial \xi}(\xi_N))^3}}{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_N))^2} - \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_N)}{(\frac{\partial S}{\partial \xi}(\xi_N))^3}} \tag{32}
$$

$$
\bar{r}_N = \frac{\frac{2}{(\frac{\partial S}{\partial \xi}(\xi_N))^2}}{\frac{1}{(\frac{\partial S}{\partial \xi}(\xi_N))^2} - \frac{\Delta\xi}{2}\frac{\frac{\partial^2 S}{\partial \xi^2}(\xi_N)}{(\frac{\partial S}{\partial \xi}(\xi_N))^3}} \tag{33}
$$

The last two boundary conditions correspond to the missing atomic tridiagonal matrices at the top left and bottom right of the tridiagonal matrix.

For the boundary condition corresponding to j = M+1, we have

$$
\begin{aligned}
-a_{i,M}U^{k+1}_{i-1,M-1} - b_{i,M}U^{k+1}_{i,M-1} + a_{i,M}U^{k+1}_{i+1,M-1} - c_{i,M}U^{k+1}_{i-1,M} + d_{i,M}U^{k+1}_{i,M} - e_{i,M}U^{k+1}_{i+1,M} \\
= U^k_{i,M} - a_{i,M}S(\xi_{i-1}) + f_{i,M}S(\xi_i) + a_{i,j}S(\xi_{i+1})
\end{aligned} \tag{34}
$$

And for the boundary condition corresponding to j = 1, we have

$$
\begin{aligned}
-c_{i,2}U^{k+1}_{i-1,2} + d_{i,2}U^{k+1}_{i,2} - e_{i,2}U^{k+1}_{i+1,2} + a_{i,2}U^{k+1}_{i-1,3} - f_{i,2}U^{k+1}_{i,3} - a_{i,2}U^{k+1}_{i+1,3} \\
= U^k_{i,j} + a_{i,2}U^{k+1}_{i-1,1} + b_{i,2}U^{k+1}_{i,1} - a_{i,2}U^{k+1}_{i+1,1}
\end{aligned} \tag{35}
$$

Why a Numerical Heston PDE scheme?

The goal of our project is to price SPY European options. The advantages of Numerical methods are that they are often easy to use and can produce results quickly. Additionally, they can solve equations where an analytic solution is impossible. In our project, after we address all the boundary conditions by using the method above, we can solve the linear equation at each time step to compute a numerical solution. This fully implicit method is unconditionally stable.

Results and Conclusions

In order to provide an example, we calculate the price grid using the Euro call option on SPY with maturity=0.5, risk-free rate=0.01, dividend yield=0, K=4552.48 (the close price of SPY on 4/1/2022), The following are the results we obtained from our calibration.

Table 1. Heston Model Parameters

kappa	theta	sigma	rho	v0
2	0.07844994	0.74456024	-0.82445271	0.0283379

The bounds used in this calibration were (0.0, 2.0), (0.0, 1.0), (0.001, 5.0), (-1, 1), (0.0, 2.0) and our initial guesses were 0.25, 0.1, 1.5, -0.5, 0.2 for kappa, theta, sigma, rho, and v0 respectively. Our bounds for calibration were obtained from our homework. With these parameters, we applied our code for pricing with the Heston PDE to create the following volatility skews.

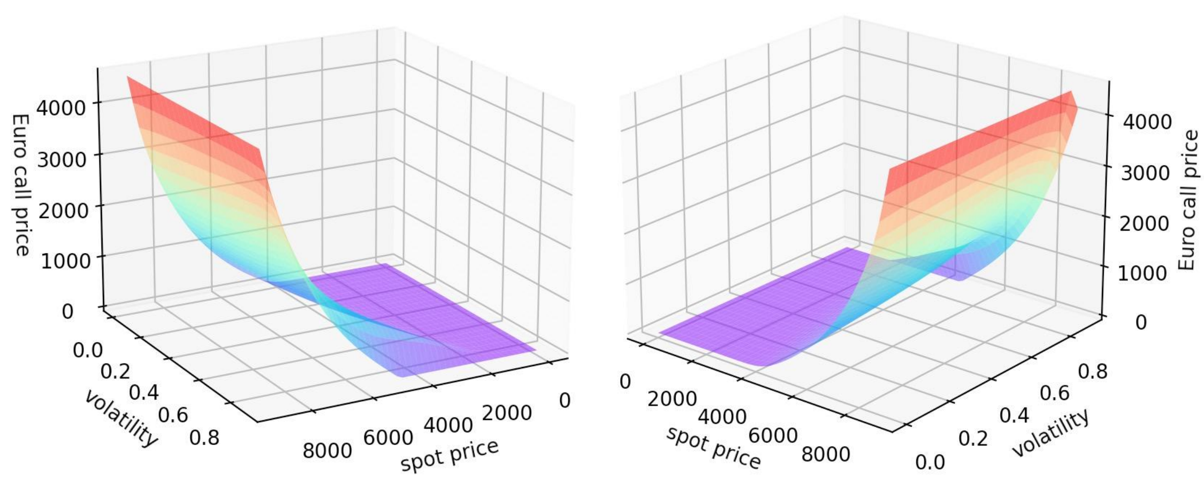


Figure 1. Two Sides of View of Volatility Skews

Comparison with Market Prices

For the last part of our project, after we obtain our prices, we wanted to check our answers with the market data. We compared our results prices with the market price from the data we used to calibrate. We compared the market price and our model price for a given strike and volatility combination. Below are tables of our obtained market prices, our obtained model prices, and the difference between market and model prices, respectively. As you can see, the differences are larger for moneyness values closer to 100.

Market Prices

Moneyness	1M	2M	3M	4M	5M	6M
75	1139.28	1139.94	1146.25	1158.18	1166.69	1177.92
80	912.87	916.71	927.58	944.07	956.77	971.65
85	687.57	697.46	715.26	737.49	755.16	773.78
90	465.45	486.9	513.55	541.65	564.21	586.34
95	255.74	292.29	326.74	359.92	386.97	412.42
100	82.08	126.47	163.82	199.5	229.36	256.68
105	6.04	26.89	51.95	79.65	105.35	129.9
110	0.3	3.35	11.06	22.38	35.58	50.59
115	0.07	0.63	2.36	5.95	10.82	17.36
120	0.03	0.29	0.8	1.89	3.72	6.47
125	0.02	0.15	0.39	0.89	1.58	2.75

Table 2. Moneyness of Options

Model Prices: Rows = strikes, columns = tenors in months

	1	2	3	4	5	6
75	1161.609193	1154.312953	1228.501244	1273.369705	1301.090434	1328.925626
80	950.059932	991.978163	1042.272231	1089.052987	1121.701584	1152.381780
85	743.403727	802.930768	860.221210	910.520942	947.410475	980.443773
90	546.449801	623.048369	686.463697	739.396113	779.228103	813.998748
95	364.946353	452.479834	518.932547	573.715680	615.981888	652.531771
100	203.002795	291.728261	357.871651	413.759663	458.132172	496.346086
105	81.077120	153.962960	213.298303	265.563828	309.167930	347.696433
110	32.045743	73.245586	114.612242	153.175841	188.313454	222.121461
115	15.641983	36.343496	59.164476	84.212487	108.152716	133.257374
120	7.895883	20.834786	32.022512	45.586782	60.898408	78.071338
125	4.501163	12.061004	18.518890	26.592996	34.726721	45.378056

Absolute difference between Model and Market: Rows = strikes, columns = tenors in months

	1	2	3	4	5	6
75	22.329193	14.372953	82.251244	115.189705	134.400434	151.005626
80	37.189932	75.268163	114.692231	144.982987	164.931584	180.731780
85	55.833727	105.470768	144.961210	173.030942	192.250475	206.663773
90	80.999801	136.148369	172.913697	197.746113	215.018103	227.658748
95	109.206353	160.189834	192.192547	213.795680	229.011888	240.111771
100	120.922795	165.258261	194.051651	214.259663	228.772172	239.666086
105	75.037120	127.072960	161.348303	185.913828	203.817930	217.796433
110	31.745743	69.895586	103.552242	130.795841	152.733454	171.531461
115	15.571983	35.713496	56.804476	78.262487	97.332716	115.897374
120	7.865883	20.544786	31.222512	43.696782	57.178408	71.601338
125	4.481163	11.911004	18.128890	25.702996	33.146721	42.628056

References

1. *Hirsa, A. (2012), Computational Methods in Finance-CRC Press*