**Problem Set # 3**

**1. Implementation of Breeden-Litzenberger:**

**(a)** According to the BS formular, I got the strike prices below:

一些文字和图案

描述已自动生成

**(b)**Using np.polyfit, I got the volatility function for all strikes, *σ*(*K*):

图表, 折线图

描述已自动生成

**(c)** Here’s the plot the neutral density:

图表, 折线图

描述已自动生成

**(d)** Using a constant volatility, I can get the new density plot below:

图表, 折线图

描述已自动生成

**(e):**

(i) 1M European Digital Put Option with Strike 110 is about: **1.00**

(ii) 3M European Digital Call Option with Strike 105 is about: **0.36**

(iii) 2M European Call Option with Strike 100 is about: **2.76**

**2. Calibration of Heston Model:**

**(a)** There’re three main methods to detect the arbitrage. First, call(put) prices are monotonically decreasing(increasing). Second, call(put) prices whose rate of change is greater than 0 (-1) and less than 1(0). Third, Call and put prices that are convex with respect to changes in strike.

After checking, the options have no chances for arbitrage.

**(b)(c)** In a form of **[﻿kappa, theta, sigma, rho, v0]**, by changing bounds and start value, I got the result below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Initial Value** | **Lower bound** | **Upper bound** | **Final Value** | **Least Squares** |
| [0,0.2,0.2,0,0.2] | ﻿[0.01, 0.01, 0.0, -1, 0] | ﻿[2.5, 1, 1, 0.5, 0.5] | ﻿[ 1.26997742, 0.08847748, 1, -0.82384662, 0.03426619] | ﻿33.494880284930886 |
| ﻿[0.2, 0.3, 0.3, 0.1, 0.3] | ﻿[0.01, 0.01, 0.0, -1, 0] | ﻿[2.5, 1, 1, 0.5, 0.5] | ﻿[ 1.27005383, 0.08847459, 1, -0.82384995, 0.03426606] | ﻿33.49488031669151 |
| ﻿[0.2, 0.3, 0.3, 0.1, 0.3] | ﻿[0.001, 0, 0.0, -1, 0] | ﻿[5, 2, 2, 1, 1] | ﻿[ 4.1604416, 0.0590171, 1.67455619, -0.809526, 0.03980849] | ﻿32.17558690782599 |

From this form, we can conclude that the least squares nearly have significant connection with the bounds and the start value because it is around 33. For the final value, they vary from with each other.

All of these parameters seem not to be influenced by start value. However, kappa and theta would be influenced by the bounds apparently. For example, we can see that by changing upper bound from 2.5 to 5, kappa changes from 1.27 to 4.16 , that’s not a small shift comparing to other parameters.

**(d)** By using the unequal function, repeat what I did above and get the form below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Initial Value** | **Lower bound** | **Upper bound** | **Final Value** | **Least Squares** |
| [0,0.2,0.2,0,0.2] | ﻿[0.01, 0.01, 0.0, -1, 0] | ﻿[2.5, 1, 1, 0.5, 0.5] | ﻿﻿[ 2.43971485 ,0.05886434 ,1, -0.78507201,0.03284587] | ﻿﻿253.5842246586348 |
| ﻿[0.2, 0.3, 0.3, 0.1, 0.3] | ﻿[0.01, 0.01, 0.0, -1, 0] | ﻿[2.5, 1, 1, 0.5, 0.5] | ﻿﻿[ 2.43966924, 0.0588646, 1, -0.78507271,0.03284602] | ﻿﻿253.58422485269378 |
| ﻿[0.2, 0.3, 0.3, 0.1, 0.3] | ﻿[0.001, 0, 0.0, -1, 0] | ﻿[5, 2, 2, 1, 1] | ﻿﻿[ 3.33259696, 0.05420142, 1.16166716, -0.77654707, 0.03376776] | ﻿﻿251.62993375165848 |

Using the unequal weighted function, we can see that the least squares results are much larger than the previous ones. But it still stay in stable, around 250. For the final value, there’re some small changes in values but still have the same tendency as we talked in (c).

3. **Hedging Under Heston Model:**

I use the parameters calibrated in problem 2.b to calculate.

文本, 聊天或短信

描述已自动生成

**(a)(i)** Delta of models:

Delta computed by Heston: **﻿0.47967255696321764**

Delta computed by BS: ﻿**0.3412312268682058**

Obviously, they’re not same. Heston model take the volatility effect in to consideration while BS model take the volatility constant. So I think Heston Model calculated through FFT is better.

**(ii)**To reach delta neutral, we need to short 0.48 unit of underlying asset.

**(b)(i)** Vega of models: ﻿

Vega computed by Heston: **131.79138380102174**

Vega computed by BS: **45.57996112892058**

**(ii)**They’re different. Heston model take the Rho into consideration which would affect Vega while BS model doesn’t.