Numerical methods for the Heston model

School of Applied Mathematics
A Thesis
Presented to the
Getulio Vargas Foundation

In Partial Fulfillment
of the Requirements for the Degree
M.Sc. of Applied Mathematics

Fernando O. Teixeira

setembro 19, 2017

Approved for the Division
(Mathematics)

Hugo Alexander de la Cruz Cancino

Acknowledgements

You get pseudo-order when you seek order; you only get a measure of order and control when you embrace randomness. — Nassim Nicholas Taleb

Table of Contents

List of Tables

List of Figures

Abstract

In this thesis we revisit some of the numerical methods for solving the Heston model's European call. Specifically, we approach Euler's, the Kahl-Jackel an two versions of the exact algorithm schemes. To perform this task, firstly we present a literature review which brings stochastic calculus, the Black-Scholes (BS) model and its limitations, the stochastic volatility methods and why they resolve the issues of the BS model, and the peculiarities of the numerical methods - convergence, discretization and stability. Since it is impossible to have a deep approach to all these topics, we provide recommendations when we acknowledge that the reader might need more specifics. We introduce the methods previously cited providing all our implementations in R language. Also, we deliver an R package with these functions and others.

Keywords: Heston, Stochastic, Volatility, Black-Scholes, European call, R

Chapter 1

altadvisor: ‘Your Other Advisor’

Chapter 2

This chunk ensures that the thesisdown package is

Chapter 3

The Heston Model Implementation

Chapter 4

Results

We present here the results of all the implementations that were disclosed in the previous section. We perform numerical comparisons between all the methods, setting out differences accross number of simulations and timesteps.

Heston [1] gives a closed form used for comparison as the 'true' option value and enabling the results to be exposed in terms of bias[1] and RMSE (root mean square error).[2]

The simulaton experiments were performed on a notebook with an Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz processor and 8GB of RAM running on a linux x86_64 based OS, Fedora 25. Codes were all written in R an run on version 3.4.1 "Single Candle" [2].

First of all, we chose a parametrization based on what we saw in other works, made some adjustments, like reducing the options' time to maturity due to the slower nature of R language and compared initial the results with the true option price given by the function *callHestoncf* belonging to the package NMOF [3]. Parameters can be seen in Table below.

Variables	Values
dt	0.05
k	2.00
r	0.05
rho	-0.30
S	100.00
sigma	0.20
t	0.00
tau	1.00
theta	0.09
v	0.09
X	100.00

Table 4.1: Model Parameters

[1] Defined as $\mathbb{E}\left[\hat{\alpha} - \alpha\right]$

[2] Defined as $\sqrt{\mathbb{E}((\hat{\theta} - \theta)^2)}$

To perform our simulations, we fixed a seed and saved the results in Table 4.4. Since the value given by the *callHestoncf* function with the parameters in Table 4.1 is 14.176, the method that best approached the "true" value was the modified (drift interpolated) exact algorithm with $100,000$ simulations. Altough the Euler scheme gives the same result (14.16) as the modified EA with $10,000$ simulations, it moves away from the closed form value when we run $100,000$ simulations. Results are

Simulations	Euler	KJ	EA-BK	EA-DI
1,000	13.28	13.05	14.71	14.74
10,000	14.16	13.86	14.45	14.38
100,000	14.11	13.83	14.21	14.16

Note: Simulations performed with 20 steps, except the EA BK

Table 4.2: Results

observable in Figure 4.1 also. The plot gives a good sense of possible biases associated with each method. To verify if in fact these methods present bias, we performed ten thousand simulations of ten thousand paths to the Euler, Kahl-Jackel (KJ) and drift interpolated exact algorithm (EA-DI) with 20 steps ($dt = 0.05$) . Clearly, from Figure 4.1, all three implementations produce bias (true option value is the black vertical line). Since this setup isn't ideal, we increased the steps to 100 ($dt = 0.01$) to see if our methods would converge. The methods histogram can be viewed in Figure 4.3.

Simulations	Euler	KJ	EA-BK	EA-DI
1,000	0.01	0.01	58.56	0.02
10,000	0.05	0.05	572.83	0.04
100,000	0.46	0.45	5704.57	0.34

Note: Simulations performed with 20 steps, except the EA BK

Table 4.3: Computing time (sec.)

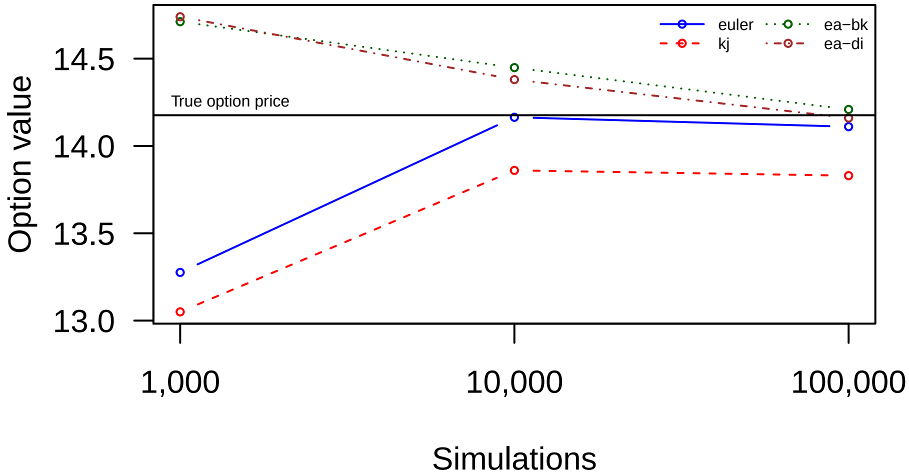

Figure 4.1: Comparison between models 20 steps

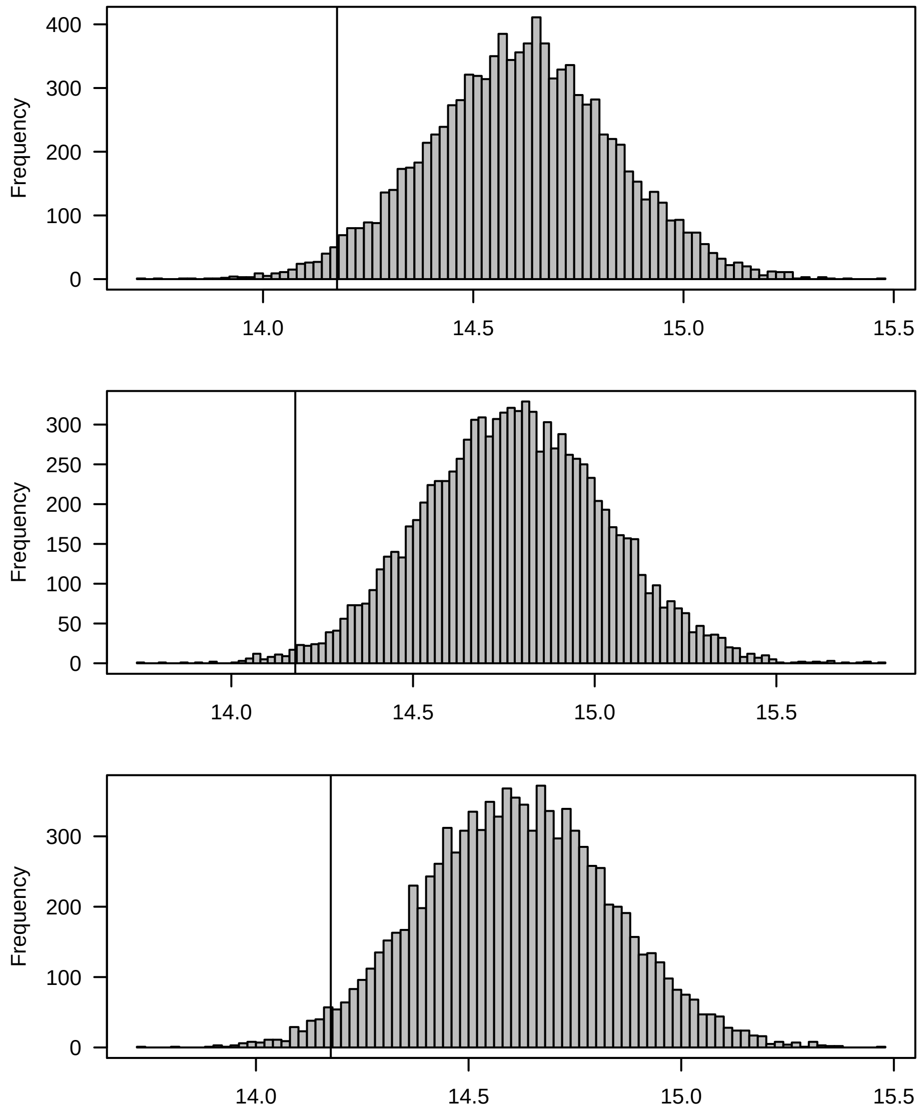

Figure 4.2: Comparison between models 20 steps

	Euler	KJ	EA-BK	EA-DI
bias	0.44	0.60	0.00	0.44
sd	0.22	0.25	0.00	0.22
RMSE	0.49	0.65	0.00	0.49
time	1.87	15.59	0.00	13.27

Note: Simulations performed with 20 steps, except the EA BK

Table 4.4: Results

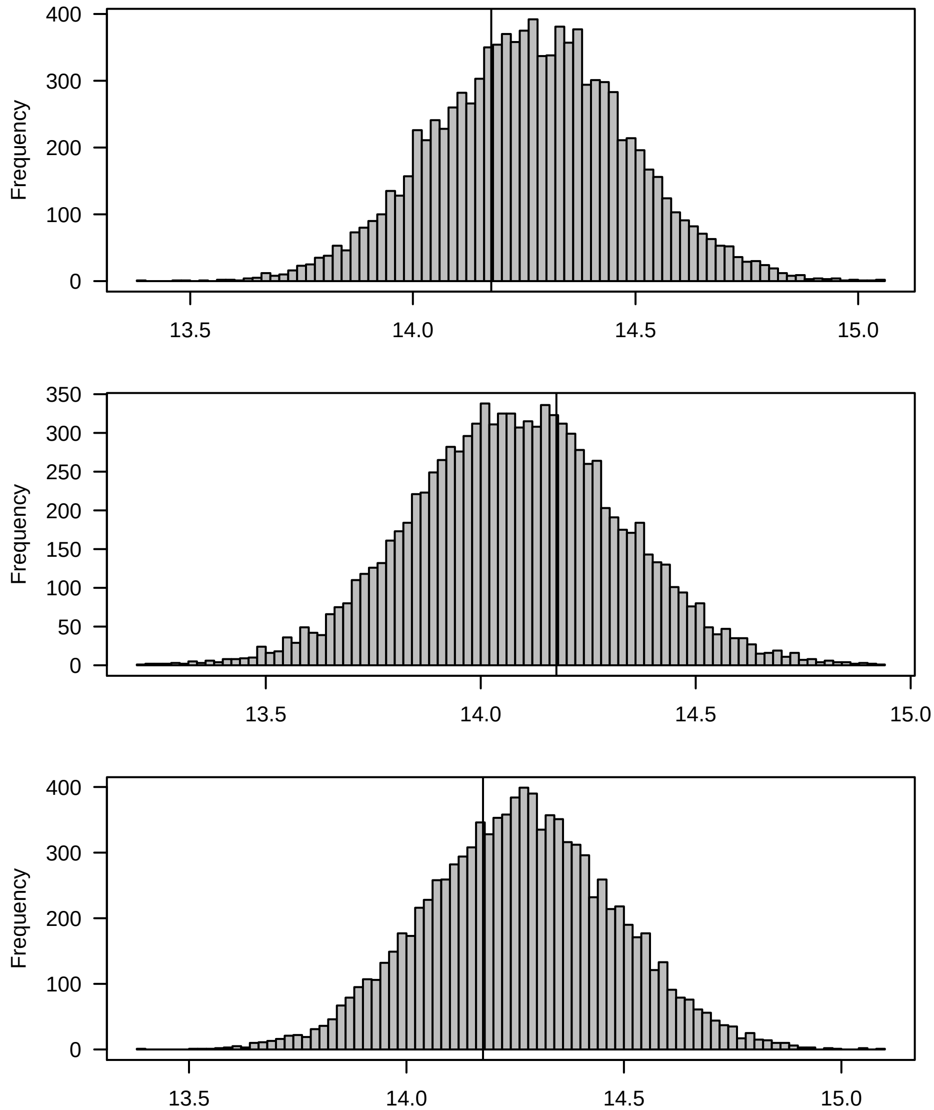

Figure 4.3: Comparison between models 100 steps

Conclusion

Chapter 5

Black-Scholes formula

References

[1] S.L. Heston, Review of Financial Studies 6 (1993) 327–343.

[2] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2017.

[3] E. Schumann, Numerical Methods and Optimization in Finance (Nmof) Manual. Package Version 0.40-0), 2011–2016.