

Advanced Computational Finance exam: Report on Exercise 10.2

Viktoriia Egorova
(879882)

26.06.2020

Formulation of the problem

The exercise consists of pricing the Asian call and put options, four barrier options (such as Down-and-Out call, Up-and-Out put, Down-and-In put, Up-and-In call) and Double Barrier option. All the payoff should be performed with Variance Gamma and Heston model.

Algorithm

The general algorithm is the following:

1. Set the parameters (either default or given from the command line). In particular, the term structure is imported from the file "ir.py".
2. Generate asset price trajectories depending on the model up to time T (which is the initial parameter, equals 1 year by default according to the task) using the Monte Carlo approach.
3. Check the martingale property of the generated trajectories in order to know if the computations can be reliable.
4. Depending of the type of the option compute the payoffs needed.
5. Compute the price of needed options and the corresponding error.

Also there are two separate files ("Heston checking.py", "VG checking.py") for testing the martingale property and reliability of the Monte Carlo simulations for option pricing for different maturities using fixed interest rates for two different cases. This is done in order to check the results with benchmark results and to understand if the results of computation could be considered as relevant.

Models

Heston model

We have

$$\begin{aligned} S_t &= S_0 e^{X_t} \\ dX_t &= -\frac{\nu_t}{2} dt + \sqrt{\nu_t} dW_t, \quad X_0 = 0 \\ d\nu_t &= k(\theta - \nu_t) dt + \eta \sqrt{\nu_t} dY_t, \quad \nu_0 = \sigma^2 \end{aligned}$$

where two innovation processes are correlated: $\mathbb{E}[dW_t dY_t] = \rho dt$. For simulation of the Heston process we use Euler scheme:

$$\begin{aligned} X_{n+1} &= X_n - \frac{\nu_n}{2} dt + \sqrt{\nu_n} dt (\rho \xi_Y + \sqrt{1 - \rho^2} \xi_Z) \\ \nu_n &= (\hat{\nu}_n)^+ \\ \hat{\nu}_{n+1} &= \nu_n + k(\theta - \nu_n) dt + \eta \sqrt{\nu_n} dt \xi_Y \end{aligned}$$

Euler scheme is the chosen approximation procedure. The tower property for conditional expectation allows to generate a volatility trajectory and given this trajectory generate several trajectories for the underlying. In order to have N overall trajectories it was chosen to divide it to \sqrt{N} trajectories of volatility and \sqrt{N} trajectories of the asset, but they are not need to be the same.

Variance Gamma model

The model is

$$Z_t = \theta\gamma(t) + \eta W_{\gamma(t)}, \quad \mathbb{P}(\gamma(t) \leq \lambda) = \frac{1}{\nu^{\frac{t}{\nu}} \Gamma(\frac{t}{\nu})} \int_0^\lambda dx e^{-\frac{x}{\nu}} x^{\frac{t}{\nu}-1}, \quad W_t \sim \mathcal{N}(0, 1)$$

The process

$$\frac{S_t}{B_t} = \exp\left(\frac{t}{\nu} \log\left(1 - \nu\theta + \frac{\nu\eta^2}{2}\right) + Z_t\right)$$

is the martingale process under the risk neutral measure. The numerical procedure is the following

$$Z_n = \theta\gamma(\delta t_n) + \eta W_{\gamma(\delta t_n)}, \quad \delta t_n = t_n - t_{n-1}, \quad \gamma(\delta t_n) \sim \Gamma\left(\frac{\delta t_n}{\nu}, \nu\right)$$

Asian option

An Asian option is a path dependent option. That is the option where the payoff depends on the average price of the underlying asset over a certain period of time, as opposed to standard options (American and European) where the payoff depends on the price of the underlying asset at a specific point in time (maturity). Due to the averaging mechanism, Asian options have relatively low volatility. Asian options are also known as average options. So

$$S(T) = \frac{1}{T} \int_0^T S(t) dt \tag{1}$$

Table 1: Default parameters for different models

Option parameters		VG		Heston	
S_0	1	η	0.1664	κ	0.1153
strike	1.03	θ	-0.7678	ν_0	0.0635
T	1	ν	0.0622	ρ	0.2125
dt	1/52			η	0.0100
J	2 ²²			θ	0.0240

Following the general algorithm we do the following:

1. Set the initial parameters :
 - maturity of the option is $T = 1$ year, with weekly time step $\Delta t = \frac{1}{52}$ and number of steps $N = 52$ since the average should be computed at the end of each week
 - number of trajectories is $J = 2^{22}$
 - strike $K = 1.03$
 - initial value of the underlying $S_0 = 1$
2. Generate trajectories of asset price with Variance Gamma/Heston model

$$S^j(t_n), \quad 0 \leq j < J, \quad 0 \leq n \leq N.$$

After this we have an array with $N + 1$ rows and J columns.

3. Check the martingale property. In order to do this compute for every time step t_n the expected value $\mathbb{E}[S(t_n)]$ which should be equal to S_0 . From the tables 7 and 8 in Appendix 1 we see that the martingale property is satisfied for both VG and Heston models.
4. For the asian option the underlying value at maturity (1) for every trajectory is computed as average of values at the end of each week:

$$S_{mc}^j(T) = \frac{1}{N+1} \sum_{n=0}^N S^j(t_n).$$

Thus the payoff for every trajectory is $\max(S_{mc}^j(T) - K, 0)$ for call and $\max(K - S_{mc}^j(T), 0)$ for put.

5. Finally, compute the Asian option value

$$C_A = \mathbb{E}[e^{-\int_0^T r(s)ds}(S(T) - K)^+], \quad P_A = \mathbb{E}[e^{-\int_0^T r(s)ds}(K - S(T))^+]$$

So numerically it will be computed as:

$$C_A \simeq \frac{1}{J} \sum_{j=0}^{J-1} P(0, T) \max(S_{mc}^j(T) - K, 0), \quad P_A \simeq \frac{1}{J} \sum_{j=0}^{J-1} P(0, T) \max(K - S_{mc}^j(T), 0)$$

The MC error is estimated as

$$Err_{call} \simeq \sqrt{\frac{\nu_{call} - C_A^2}{J}}, \quad Err_{put} \simeq \sqrt{\frac{\nu_{put} - P_A^2}{J}} \quad (2)$$

where

$$\nu_{call} \simeq \frac{1}{J} \sum_{j=0}^{J-1} P(0, T) \max(S_{mc}^j(T) - K, 0)^2, \quad \nu_{put} \simeq \frac{1}{J} \sum_{j=0}^{J-1} P(0, T) \max(K - S_{mc}^j(T), 0)^2.$$

The results of the implementation depending on the number of trajectories are presented in tables 2 and 3.

Table 2: Asian call price and MC error depending on J

J	VG as. call price	VG error	Heston as. call price	Heston error
10	0.0513642	0.0023921	0.0523	0.00277
12	0.0519745	0.0012129	0.05309	0.00142
14	0.0514311	0.0006077	0.05365	0.00071
16	0.051198	0.0003036	0.05398	0.00036
18	0.0514236	0.0001524	0.05335	0.00018
20	0.0516126	7.62e-05	0.0536	9e-05
22	0.0515242	3.81e-05	0.05353	4e-05

Table 3: Asian put price and MC error depending on J

J	VG as. put price	VG error	Heston as. put price	Heston error
10	0.06171	0.0027608	0.06381	0.00257
12	0.0630983	0.001365	0.06424	0.00126
14	0.0613545	0.0006746	0.06484	0.00064
16	0.0627054	0.0003413	0.06474	0.00032
18	0.0626505	0.000171	0.06479	0.00016
20	0.0626337	8.57e-05	0.06464	8e-05
22	0.0626515	4.28e-05	0.06464	4e-05

Since the arithmetic means does not follow lognormal distribution, there is no closed form analytical solution for arithmetic averaging Asian options. So we cannot compare numerical results with analytical results. In general (but not always), Asian options are less expensive than their standard counterparts, as the volatility of the average price is less than the volatility of the spot price.

In particular, the Asian call option price can be upper bounded by the corresponding european call option price using convexity arguments.

Proposition. Assume that $r \geq 0$, and let ϕ be a convex and nondecreasing payoff function. We have the bound

$$e^{-rT} \mathbb{E} \left[\phi \left(\frac{1}{T} \int_0^T S_u du - K \right) \right] \leq e^{-rT} E[\phi(S_T - K)].$$

Proof. By Jensen's inequality for the uniform measure with probability density function $(1/T)1_{[0,T]}$ on $[0, T]$ and for

the probability measure \mathbb{P} , we have

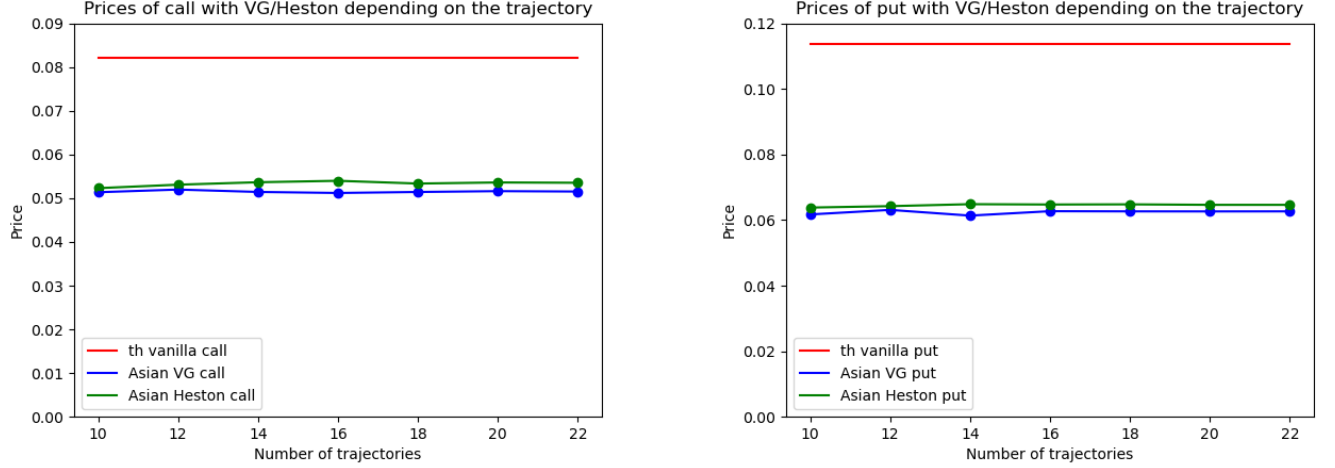
$$\begin{aligned}
e^{-rT} \mathbb{E} \left[\phi \left(\int_0^T S_u \frac{du}{T} - K \right) \right] &= e^{-rT} \mathbb{E} \left[\phi \left(\int_0^T (S_u - K) \frac{du}{T} \right) \right] \leq e^{-rT} \mathbb{E} \left[\int_0^T \phi(S_u - K) \frac{du}{T} \right] = \\
&= e^{-rT} \mathbb{E} \left[\int_0^T \phi \left(e^{-(T-u)r} \mathbb{E}[S_T | \mathcal{F}_u] - K \right) \frac{du}{T} \right] = e^{-rT} \mathbb{E} \left[\int_0^T \phi \left(\mathbb{E}[e^{-(T-u)r} S_T - K | \mathcal{F}_u] \right) \frac{du}{T} \right] \stackrel{(1)}{\leq} \\
&\stackrel{(1)}{\leq} e^{-rT} \mathbb{E} \left[\int_0^T \mathbb{E} \left[\phi \left(e^{-(T-u)r} S_T - K \right) | \mathcal{F}_u \right] \frac{du}{T} \right] \stackrel{(2)}{\leq} \\
&\stackrel{(2)}{\leq} e^{-rT} \int_0^T \mathbb{E}[\mathbb{E}[\phi(S_T - K) | \mathcal{F}_u]] \frac{du}{T} = e^{-rT} \int_0^T \mathbb{E}[\phi(S_T - K)] \frac{du}{T} = e^{-rT} \mathbb{E}[\phi(S_T - K)],
\end{aligned}$$

where from (1) to (2) were used the facts that $r \geq 0$ and ϕ is nondecreasing. \square

Taking the maximum function this proposition shows that Asian option prices are upper bounded by European call options prices.

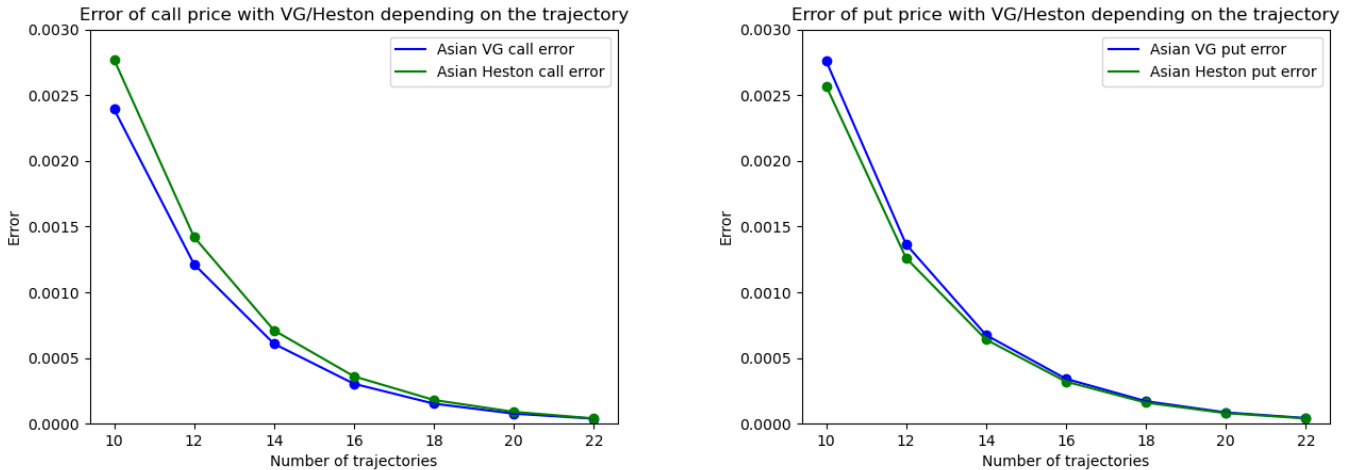
The overall results are represented on the following plots. On the plot 1 are represented the prices for the call (left) and put (right) Asian options depending on the number of trajectories. For both cases green line corresponds to Heston model and blue line corresponds to VG model. Red lines correspond to the theoretical price of vanilla option.

Figure 1: Options' prices



As we can see from the these plots, the upper bound property tends to be satisfied in both (call/put) cases. On the plot 2 are represented the errors computed by formula (2) for call (left) and put (right) Asian options. As we can see, the error for both VG and Heston model decreases significantly with increasing number of trajectories

Figure 2: Errors for options' prices



Barrier option

A barrier option is a type of derivative where the payoff depends on whether or not the underlying asset has reached or exceeded a predetermined price. A barrier option can be a knock-out, meaning it expires worthless if the underlying exceeds a certain price. Knock-out options limit losses, but also potential profits. It can also be a knock-in, meaning it has no value until the underlying reaches a certain price. Barrier options are also considered a type of path-dependent option because a barrier option's payoff is based on the underlying asset's price path.

Barrier options:

- Knock-In.

An option contract that begins to function as a normal option only if a certain price level is reached before expiration.

- Up-and-In.

The option only comes into existence if the price of the underlying asset rises above the pre-specified barrier, which is set above the underlying's initial price.

- Down-and-In.

The option only comes into existence if the price of the underlying asset rises below the pre-specified barrier, which is set above the underlying's initial price.

- Knock-Out.

An option that expires worthless if a specified price level in the underlying asset is reached.

- Up-and-Out.

The option does not exist when the underlying security moves above a barrier that is set above the underlying's initial price.

- Down-and-Out.

The option does not exist when the underlying asset moves below a barrier that is set below the underlying's initial price.

The difference between a knock-in and knock-out option is that a knock-in option comes into existence only when the underlying security reaches a barrier, while a knock-out option ceases to exist when the underlying security reaches a barrier.

Define

$$\underline{S}(T) = \min S(t), \quad \bar{S}(T) = \max S(t), \quad t \in [0, T]$$

We consider the following four types of options:

1. Down-and-Out Call. The payoff is:

$$C_{do} = \begin{cases} (S(T) - K)^+, & \underline{S}(T) > \lambda \\ 0, & \underline{S}(T) < \lambda \end{cases}$$

2. Up-and-Out Put. The payoff is:

$$P_{uo} = \begin{cases} (K - S(T))^+, & \bar{S}(T) < \Lambda \\ 0, & \bar{S}(T) > \Lambda \end{cases}$$

3. Down-and-In Put. The payoff is:

$$P_{di} = \begin{cases} (K - S(T))^+, & \underline{S}(T) < \lambda \\ 0, & \underline{S}(T) > \lambda \end{cases}$$

4. Up-and-In Call. The payoff is:

$$C_{ui} = \begin{cases} (S(T) - K)^+, & \bar{S}(T) > \Lambda \\ 0, & \bar{S}(T) < \Lambda \end{cases}$$

5. Double Barrier Option. The payoff is:

$$B = \begin{cases} S(T), & \lambda < \underline{S}(T), \bar{S}(T) < \Lambda \\ 0, & \text{otherwise} \end{cases}$$

Barrier options have both a strike price and a barrier level specified. By default set the following parameters:

Table 4: Default parameters for computing barrier options

K	1.03	λ	0.7	Λ	1.15
---	------	-----------	-----	-----------	------

The results of implementation are represented in the following tables: table 5 for VG model and 6 for the Heston model.

Table 5: Prices of Barrier options with VG model

J	DO call price	error _{DO}	UO put price	error _{UO}	DI put price	error _{DI}	UI call price	error _{UI}	double barrier price	error _{double barrier}
10	0.0770638	0.0041911	0.0952566	0.0045057	0.0587325	0.0044389	0.073893	0.0042247	0.3277772	0.0139912
12	0.0832086	0.0021738	0.0950152	0.0022281	0.0580688	0.0021924	0.0795128	0.0021938	0.3229482	0.0069839
14	0.0833556	0.0010952	0.0930346	0.0010984	0.0566021	0.0010815	0.0799947	0.0011044	0.3264674	0.0035017
16	0.0819601	0.0005449	0.0968517	0.0005651	0.060121	0.0005603	0.0786961	0.0005492	0.3269964	0.0017504
18	0.0823669	0.0002719	0.0958292	0.0002807	0.0594665	0.0002779	0.0790227	0.0002741	0.3259484	0.0008753
20	0.0823659	0.000136	0.0957236	0.0001402	0.0593544	0.0001388	0.0790278	0.0001371	0.3250897	0.0004373
22	0.0823499	6.8e-05	0.0957823	7.02e-05	0.0595345	6.95e-05	0.0789738	6.85e-05	0.3254043	0.0002188

Table 6: Prices of Barrier options with Heston model

J	DO call price	error _{DO}	UO put price	error _{UO}	DI put price	error _{DI}	UI call price	error _{UI}	double barrier price	error _{double barrier}
10	0.0855372	0.0051844	0.0949779	0.0042256	0.0497168	0.003983	0.0838327	0.0052026	0.3177719	0.0136294
12	0.083578	0.0024441	0.0979644	0.0021242	0.0500023	0.0019918	0.0814305	0.0024554	0.335375	0.0069109
14	0.0846488	0.0012365	0.098879	0.0010679	0.052456	0.0010164	0.0822668	0.0012427	0.3342068	0.003461
16	0.084773	0.0006205	0.0994291	0.0005364	0.0528269	0.000511	0.0825621	0.0006234	0.3295804	0.0017219
18	0.0837478	0.0003074	0.0993596	0.0002685	0.052955	0.0002557	0.0815243	0.0003089	0.3298169	0.0008616
20	0.0843274	0.0001544	0.0991078	0.0001342	0.0526969	0.0001276	0.0821189	0.0001551	0.3295868	0.0004307
22	0.0843153	7.72e-05	0.0991057	6.7e-05	0.0526845	6.37e-05	0.0821068	7.76e-05	0.03295638	0.0002154

Barrier options are cheaper than traditional vanilla options, primarily because the barrier increases the chances of the option expiring worthless (a zero payoff may occur before expiry). The following plots represent this fact in our case.

Figure 3: Down-and-Out call price and error

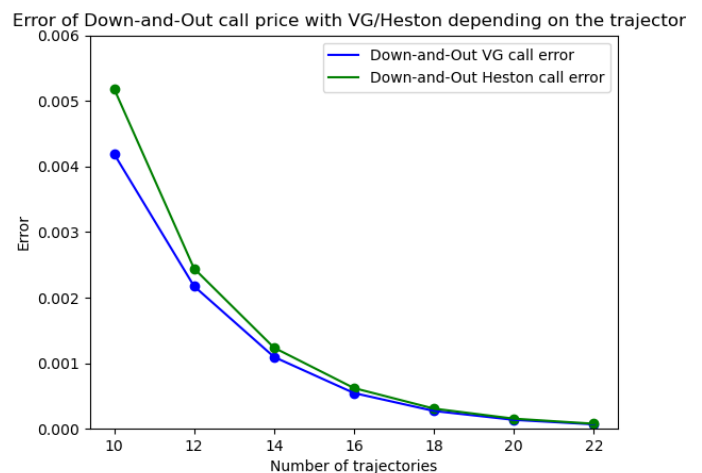
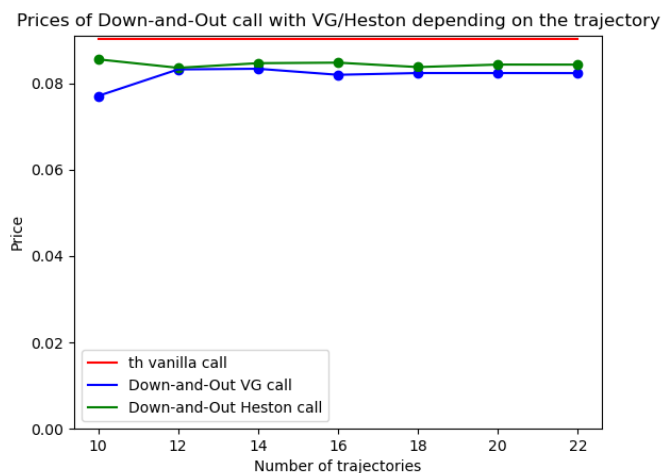


Figure 4: Up-and-Out put price and error

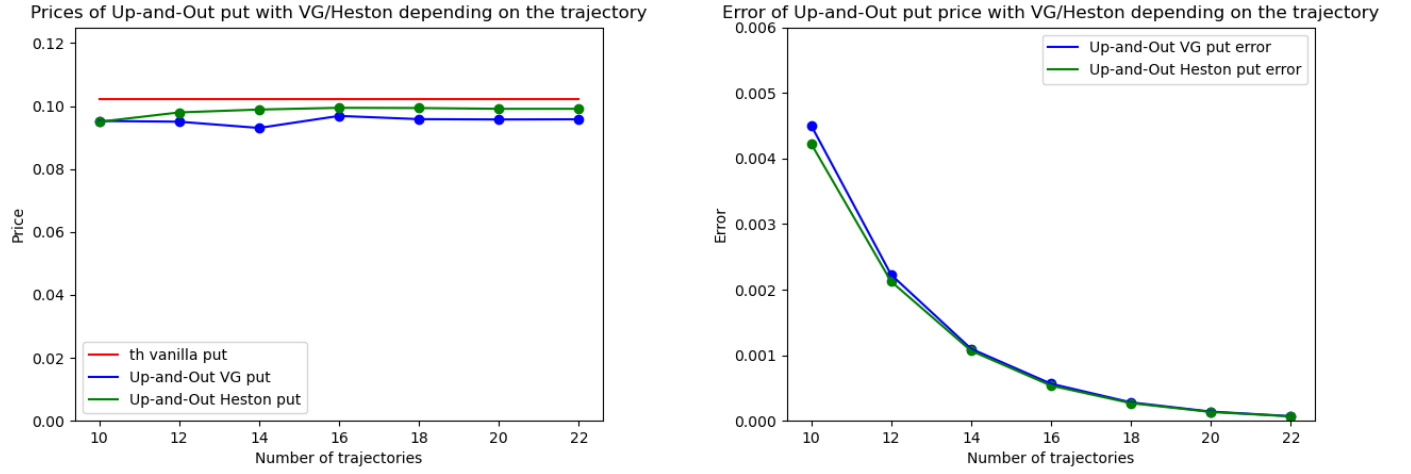


Figure 5: Down-and-In put price and error

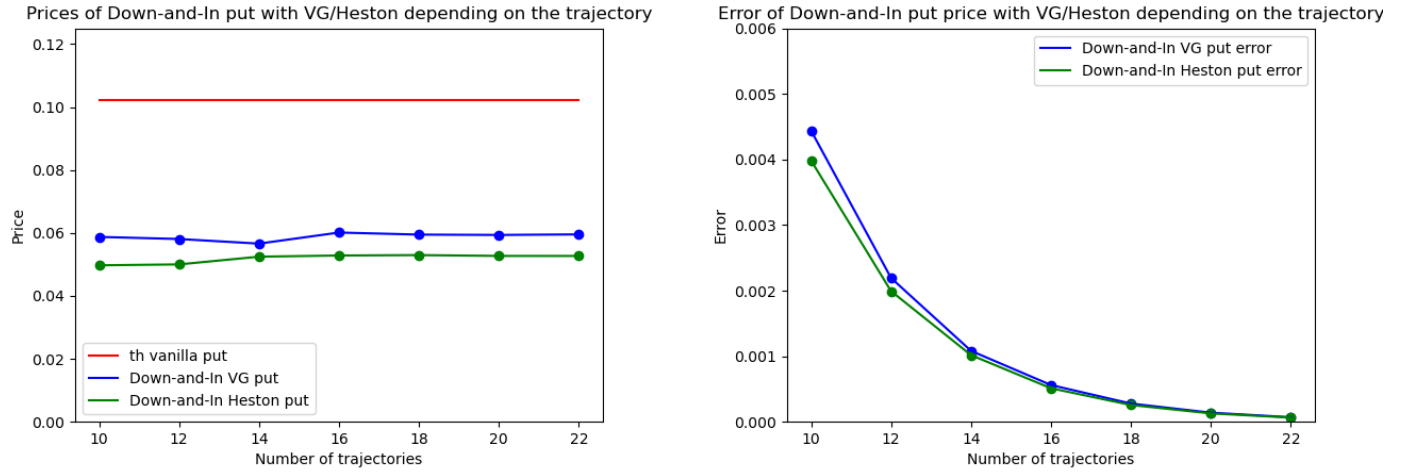
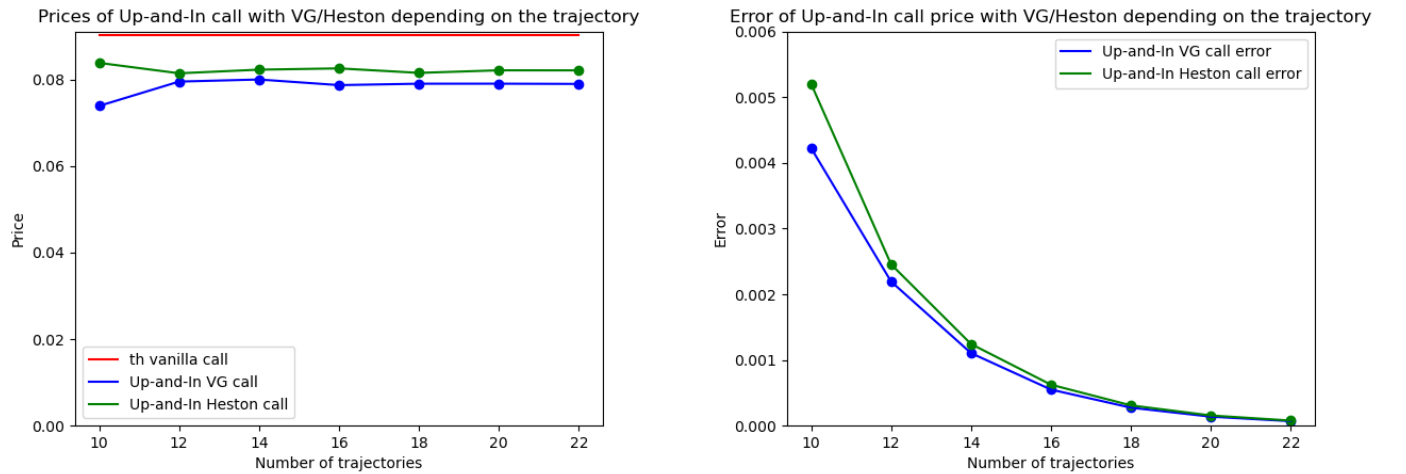


Figure 6: Up-and-In call price and error



Appendix 1. Martingale property

The results of checking of martingale property are represented in tables 7 and 8. Here absolute error is the absolute value of difference between obtained mean and S_0 and Monte Carlo error is the standard deviation of the average estimator.

Figure 7: Simulation of trajectories

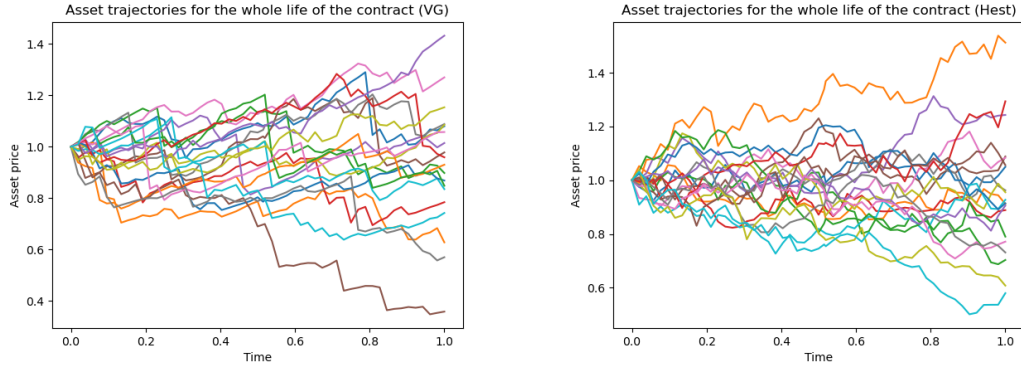


Figure 8: Standard deviation and Monte Carlo error (VG)

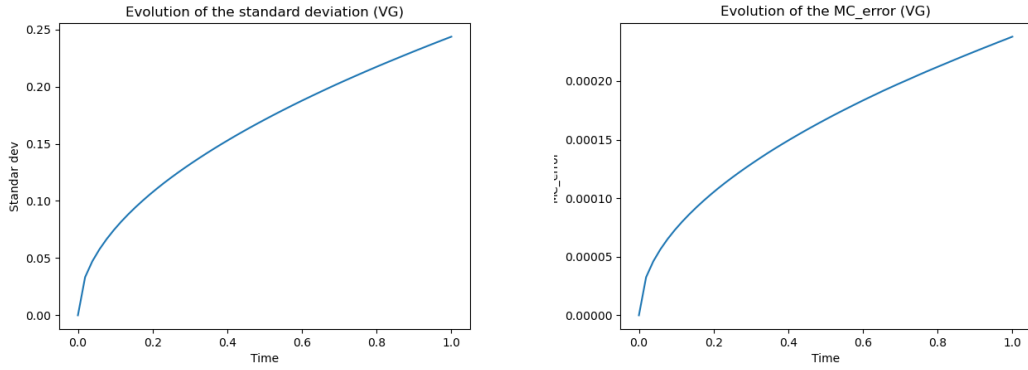


Figure 9: Standard deviation and Monte Carlo error (Heston)

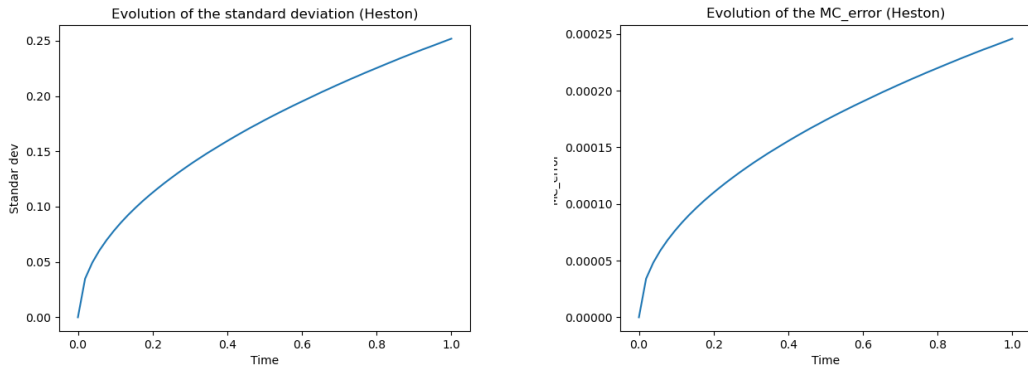


Table 7: Martingale property (VG)

week	maturity	mean	Abs error	MC error
0	0	1.0	0.0	0.000000
1	0.019231	0.999977	2.3e-05	0.000048
2	0.038462	0.999982	1.8e-05	0.000069
3	0.057692	1.000021	2.1e-05	0.000084
4	0.076923	1.000019	1.9e-05	0.000099
5	0.096154	1.000024	2.4e-05	0.000108
6	0.115385	1.000027	2.7e-05	0.000120
7	0.134615	1.000042	4.2e-05	0.000129
8	0.153846	1.00002	2e-05	0.000138
9	0.173077	0.999989	1.1e-05	0.000147
10	0.192308	0.999989	1.1e-05	0.000156
11	0.211538	0.999994	6e-06	0.000162
12	0.230769	0.999994	6e-06	0.000171
13	0.25	0.999985	1.5e-05	0.000177
14	0.269231	1.000023	2.3e-05	0.000183
15	0.288462	1.000026	2.6e-05	0.000189
16	0.307692	1.00003	3e-05	0.000195
17	0.326923	1.00005	5e-05	0.000201
18	0.346154	1.000011	1.1e-05	0.000207
19	0.365385	1.000018	1.8e-05	0.000213
20	0.384615	1.000017	1.7e-05	0.000219
21	0.403846	1.000015	1.5e-05	0.000225
22	0.423077	1.000019	1.9e-05	0.000231
23	0.442308	1.000029	2.9e-05	0.000234
24	0.461538	1.000028	2.8e-05	0.000240
25	0.480769	1.000047	4.7e-05	0.000246
26	0.5	1.000057	5.7e-05	0.000252
27	0.5147	1.000048	4.8e-05	0.000255
28	0.519231	1.000057	5.7e-05	0.000255
29	0.538462	1.000038	3.8e-05	0.000261
30	0.557692	1.00004	4e-05	0.000264
31	0.576923	1.000048	4.8e-05	0.000270
32	0.596154	1.000051	5.1e-05	0.000273
33	0.615385	1.000075	7.5e-05	0.000279
34	0.634615	1.000038	3.8e-05	0.000282
35	0.653846	1.000042	4.2e-05	0.000288
36	0.673077	1.000032	3.2e-05	0.000291
37	0.692308	1.000025	2.5e-05	0.000297
38	0.711538	1.000013	1.3e-05	0.000300
39	0.730769	1.0	0.0	0.000303
40	0.75	1.000016	1.6e-05	0.000309
41	0.769231	1.000013	1.3e-05	0.000312
42	0.788462	0.999973	2.7e-05	0.000315
43	0.807692	0.999993	7e-06	0.000321
44	0.826923	0.999978	2.2e-05	0.000324
45	0.846154	0.999956	4.4e-05	0.000327
46	0.865385	0.999965	3.5e-05	0.000330
47	0.884615	1.000014	1.4e-05	0.000336
48	0.903846	1.000034	3.4e-05	0.000339
49	0.923077	1.000057	5.7e-05	0.000342
50	0.942308	1.000067	6.7e-05	0.000345
51	0.961538	1.000033	3.3e-05	0.000351
52	0.980769	1.000034	3.4e-05	0.000354
53	1.0	1.000026	2.6e-05	0.000357

Table 8: Martingale property (Heston)

week	maturity	mean	Abs error	MC error
0	0	1.0	0.0	0.0
1	0.019231	1.000058	5.8e-05	0.0001023
2	0.038462	1.000029	2.9e-05	0.0001446
3	0.057692	1.000016	1.6e-05	0.0001773
4	0.076923	1.000038	3.8e-05	0.0002046
5	0.096154	1.000053	5.3e-05	0.0002289
6	0.115385	1.000075	7.5e-05	0.0002511
7	0.134615	1.000075	7.5e-05	0.0002712
8	0.153846	1.000092	9.2e-05	0.0002898
9	0.173077	1.000166	0.000166	0.0003075
10	0.192308	1.000167	0.000167	0.000324
11	0.211538	1.000145	0.000145	0.0003396
12	0.230769	1.000171	0.000171	0.0003546
13	0.25	1.000198	0.000198	0.000369
14	0.269231	1.000185	0.000185	0.0003831
15	0.288462	1.00013	0.00013	0.0003966
16	0.307692	1.000197	0.000197	0.0004098
17	0.326923	1.000158	0.000158	0.0004221
18	0.346154	1.000144	0.000144	0.0004344
19	0.365385	1.000106	0.000106	0.0004461
20	0.384615	1.000099	9.9e-05	0.0004578
21	0.403846	1.000136	0.000136	0.0004692
22	0.423077	1.000159	0.000159	0.0004803
23	0.442308	1.000143	0.000143	0.0004911
24	0.461538	1.000176	0.000176	0.0005016
25	0.480769	1.000164	0.000164	0.0005118
26	0.5	1.000191	0.000191	0.000522
27	0.5147	1.00016	0.00016	0.0005295
28	0.519231	1.000178	0.000178	0.0005319
29	0.538462	1.000147	0.000147	0.0005418
30	0.557692	1.000174	0.000174	0.0005511
31	0.576923	1.000164	0.000164	0.0005607
32	0.596154	1.000116	0.000116	0.0005697
33	0.615385	1.000109	0.000109	0.0005787
34	0.634615	1.000093	9.3e-05	0.0005877
35	0.653846	1.000108	0.000108	0.0005967
36	0.673077	1.000068	6.8e-05	0.0006054
37	0.692308	1.000096	9.6e-05	0.0006141
38	0.711538	1.000072	7.2e-05	0.0006225
39	0.730769	1.00012	0.00012	0.0006312
40	0.75	1.000116	0.000116	0.000639
41	0.769231	1.000089	8.9e-05	0.0006474
42	0.788462	1.000039	3.9e-05	0.0006552
43	0.807692	1.000056	5.6e-05	0.0006633
44	0.826923	1.000053	5.3e-05	0.0006711
45	0.846154	1.000075	7.5e-05	0.0006789
46	0.865385	1.000084	8.4e-05	0.0006867
47	0.884615	1.000064	6.4e-05	0.0006942
48	0.903846	1.000115	0.000115	0.0007017
49	0.923077	1.000112	0.000112	0.0007092
50	0.942308	1.000077	7.7e-05	0.0007161
51	0.961538	1.00003	3e-05	0.0007233
52	0.980769	1.000037	3.7e-05	0.0007305
53	1.0	1.000071	7.1e-05	0.0007377

Appendix 2. Benchmark results

Table 9: Benchmark results for VG

r = 0							
t	put benchmark	call benchmark	error	MC put	MC call	MC put error	MC call error
0.25	0.0636	0.0336	3.4e-04	0.063493	0.033486	0.000127	7.6e-05
0.5	0.0843	0.0543	4.4e-04	0.084473	0.054561	0.000164	0.000127
0.75	0.1003	0.0703	5.1e-04	0.100437	0.070528	0.00019	0.000168
1	0.1136	0.0836	5.6e-04	0.11381	0.083958	0.000211	0.000204
1.5	0.136	0.106	6.5e-04	0.136259	0.10639	0.000245	0.000267
2	0.155	0.125	7.3e-04	0.15505	0.125037	0.000272	0.000324
r = 0.05							
t	put benchmark	call benchmark	error	put	call	put error	call error
0.25	0.0565	0.0393	1.6e-04	0.056569	0.039357	0.000121	8.3e-05
0.5	0.0709	0.0663	2.0e-04	0.070966	0.066484	0.000151	0.00014
0.75	0.0803	0.0882	2.3e-04	0.080431	0.088432	0.000171	0.000187
1	0.873	0.1075	2.5e-04	0.087382	0.107764	0.000186	0.000229
1.5	0.097	0.1414	2.7e-04	0.097145	0.141701	0.000206	0.000304
2	0.1033	0.1713	2.9e-04	0.103482	0.171486	0.00022	0.000371

Table 10: Benchmark results for Heston

r = 0							
t	put benchmark	call benchmark	error	MC put	MC call	MC put error	MC call error
0.25	0.0668	0.0368	4.8e-04	0.067253	0.037221	0.000115	0.000101
0.5	0.0872	0.0572	6.2e-04	0.087516	0.057556	0.000152	0.000154
0.75	0.1024	0.0724	7.1e-04	0.102884	0.072999	0.000178	0.000195
1	0.1154	0.0854	7.9e-04	0.115849	0.085908	0.000199	0.000232
1.5	0.1368	0.1068	9.2e-04	0.137009	0.106976	0.000232	0.000295
2	0.1538	0.1238	1.0e-03	0.154487	0.124417	0.000257	0.00035
r = 0.05							
t	put benchmark	call benchmark	error	put	call	put error	call error
0.25	0.0592	0.042	4.4e-04	0.059598	0.042361	0.000109	0.000107
0.5	0.0726	0.068	5.5e-04	0.072937	0.068408	0.000139	0.000166
0.75	0.0811	0.089	6.1e-04	0.081549	0.089573	0.000159	0.000214
1	0.0875	0.1077	6.6e-04	0.087889	0.108181	0.000173	0.000256
1.5	0.0958	0.1403	7.3e-04	0.096077	0.140469	0.000193	0.000329
2	0.1004	0.1684	7.7e-04	0.100954	0.168901	0.000206	0.000394