Mathematical Modeling and Computation in Finance
With Exercises and Python and MATLAB Computer Codes

C.W. Oosterlee & L.A. Grzelak

Solutions to exercises from Chapter 9

`https://QuantFinanceBook.com`

Ex. 9.1. As seen in Chapter 1, the process,

$$X(t) = \int_0^t \cos(t)\mathrm{d}W(t),$$

follows, at time t, a normal distribution with mean 0 and variance equal to,

$$\mathbb{V}\mathrm{ar}[X(t)] = \int_0^t \cos^2(t)\mathrm{d}t = \frac{1}{2}t + \frac{\sin(2t)}{4}.$$

The corresponding computer simulation generates the following output,

```
E(X(T)) = -1.438849e-17  and Var(X(T))=0.804350;
Exact solution: E(X(T)) = 0.000000e+00  and Var(X(T))=0.810799.
```

The computer code under the Python icon below provides us with the plot presented in Figure 1.

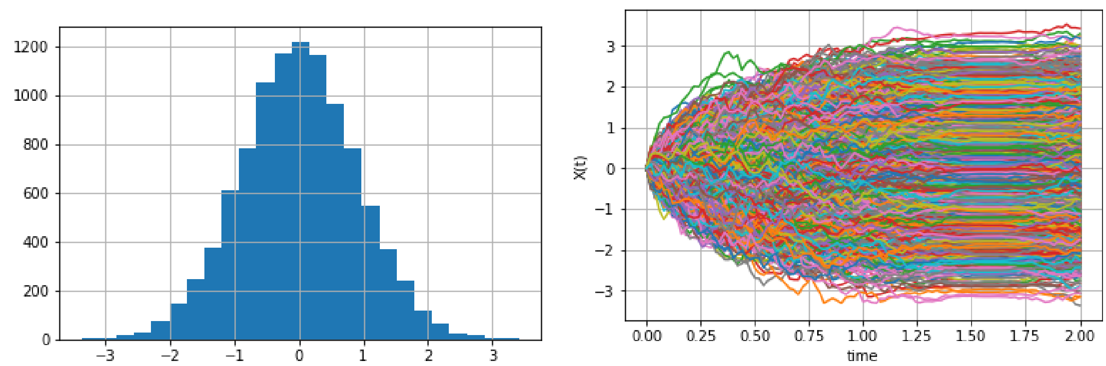

Figure 1: Results for Exercise 9.1.

Ex. 9.3. As discussed in Chapter 1, the process,

$$X(t) = \int_0^t W^4(t)\mathrm{d}W(t),$$

follows, at time t, a normal distribution with mean 0 and variance equal to,

$$\mathbb{V}\mathrm{ar}[X(t)] = \int_0^t \mathbb{E}[W^8(t)]\mathrm{d}t = 105\int_0^t t^4\mathrm{d}t = 21t^5$$

The corresponding computer simulation generates the following output,

```
E(X(T)) = 1.497488e-02  and Var(X(T))=852.556276;
Exact solution: E(X(T)) = 0.000000e+00  and Var(X(T))=672.000000.
```

The computer code below provides us with the plot in Figure 2.

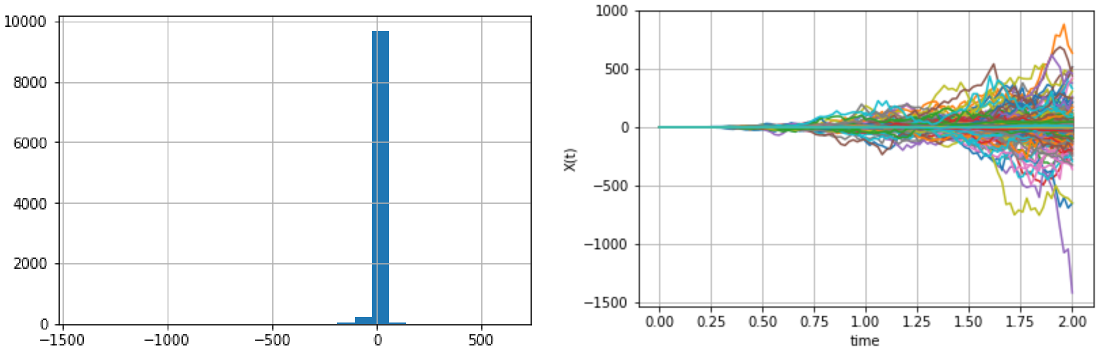

Figure 2: Results for Exercise 9.3.

Compared to Exercises 1 and 2, the convergence of the variance is much slower in this experiment.

Ex. 9.5. *a.* The computer code below provides us with the following results:

```
mean value equals to: 0.05 while the expected value is W(0) =0.00;
The error is equal to: 0.0000000000000000888.
```

b. Based on the code of Exercise a, the results are found in Figure 3

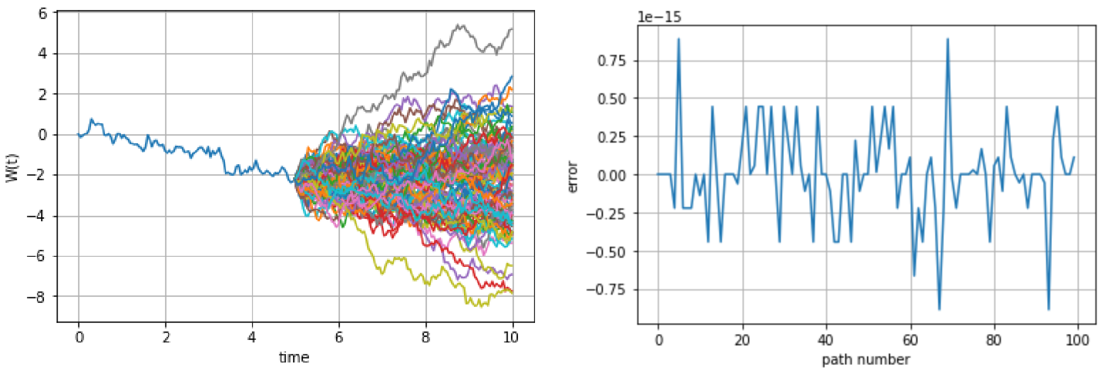

Figure 3: Results for Exercise 9.5, Left: example of nested Monte Carlo simulation; Right: error, per path.

Ex. 9.7. The expression for the variance is given by,

$$\begin{aligned}
\mathbb{V}\mathrm{ar}[X(t)] &= \mathbb{E}[X^2(t)] \\
&= \mathbb{E}[\left(W(t) - \frac{t}{T}W(T-t)\right)^2] \\
&= \mathbb{E}[W^2(t)] - 2\frac{t}{T}\mathbb{E}[W(t)W(T-t)] + \frac{t^2}{T^2}\mathbb{E}[W^2(T-t)] \\
&= t - 2\frac{t}{T}\min(t, T-t) + \frac{t^2}{T^2}(T-t).
\end{aligned}$$

The computer code below the Python icon provides us with the plot in Figure 4.

python

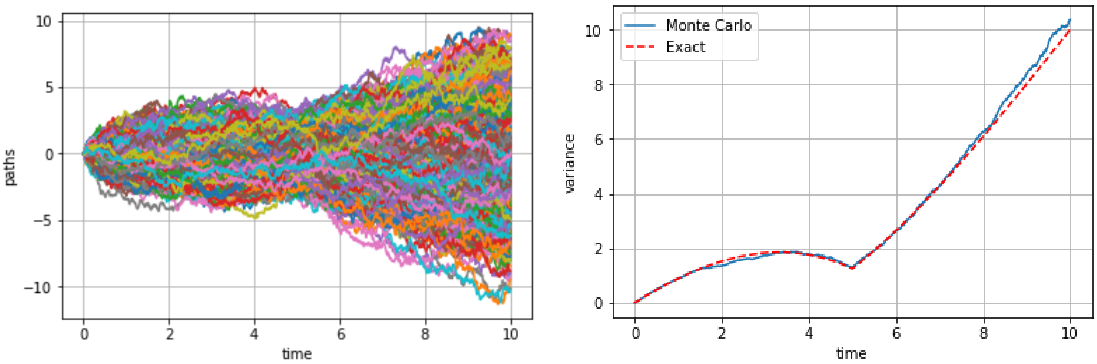

Figure 4: Results for Exercise 9.7, computation of the variance for the Brownian bridge.

Ex. 9.9. *a.* As described in Chapter 8 (before Definition 8.1.1), $v(t)$ stays positive when the Feller condition is satisfied. However, negative realizations may appear due to the discretization of the SDE. For the Euler discretization, this can be seen as follows,

$$v_{i+1} = v_i + \kappa\left(\bar{v} - v_i\right)\Delta t + \gamma\sqrt{v_i \Delta t}Z, \tag{0.1}$$

where $Z \sim \mathcal{N}(0,1)$. The righthand side of (0.1) may become negative, even when the Feller condition is satisfied. The probability of negative v-values can be assessed as follows,

$$\begin{aligned}
\mathbb{P}\left[v_{i+1} < 0\right] &= \mathbb{P}\left[v_i + \kappa\left(\bar{v} - v_i\right)\Delta t + \gamma\sqrt{v_i\Delta t}Z < 0\right] \\
&= \mathbb{P}\left[Z < \frac{-v_i - \kappa\left(\bar{v} - v_i\right)\Delta t}{\gamma\sqrt{v_i\Delta t}}\right] \\
&= F_{\mathcal{N}(0,1)}\left(\frac{-v_i - \kappa\left(\bar{v} - v_i\right)\Delta t}{\gamma\sqrt{v_i\Delta t}}\right).
\end{aligned}$$

From the expression above, we see that the likelihood of negative v_{i+1} depends on the time step Δt and the model parameters. The following computer code provides us with the plot presented in Figure 5.

python

b. The computer code below the following Python icon provides us with the plot in Figure 6.

python

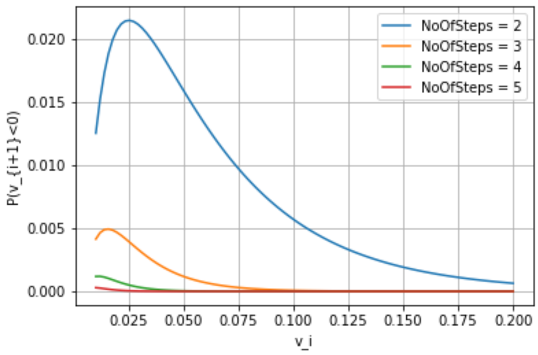

Figure 5: Results for Exercise 9.9a. $\mathbb{P}[v_{i+1} < 0]$ as a function of the previous step v_i. In this experiment we have the following parameters, $T = 1.0$, $\kappa = 0.4$, $\bar{v} = 0.1$, $\gamma = 0.25$. the Feller condition is satisfied.

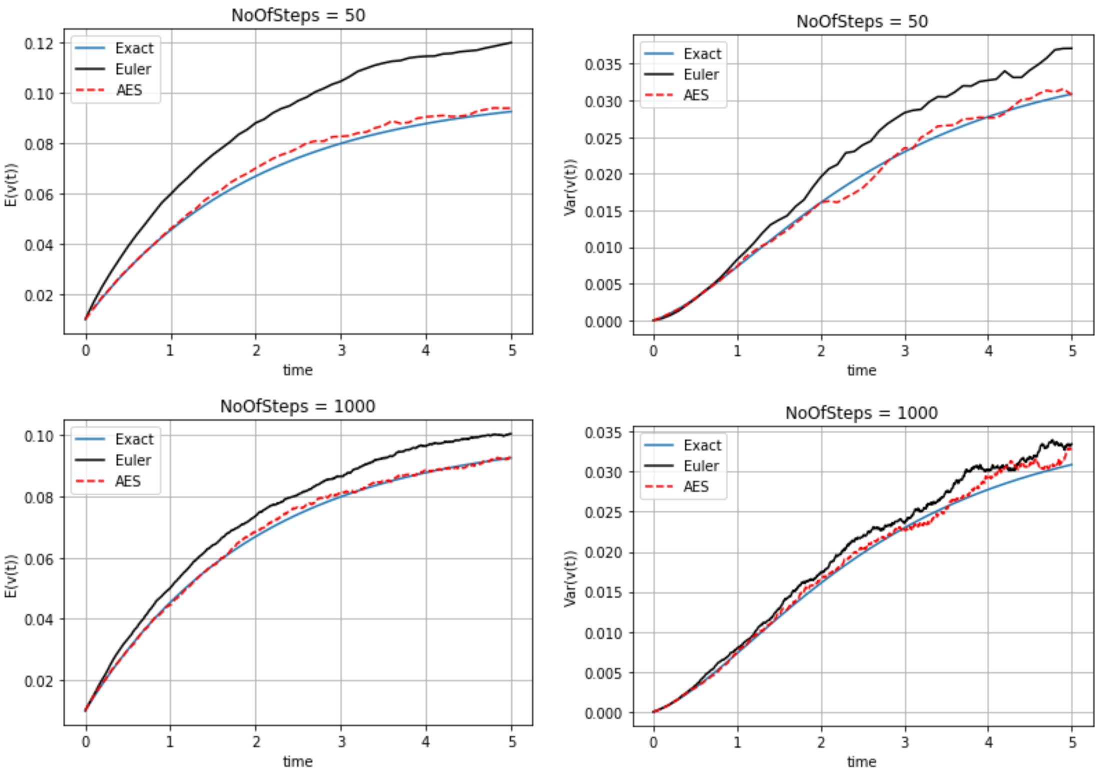

Figure 6: Results for Exercise 9.9b. Convergence of the expectation and variance depending on the number of time steps. The top row corresponds to $N = 50$, the bottom row to $N = 1000$ time steps.

Ex. 9.11. This exercise consists of three different simulations: VG, CGMY and NIG models.

a. For the VG model we follow Definition 5.4.3, where the solution for the VG model is given by:

$$
\begin{aligned}
S(t) &= S(t_0)\mathrm{e}^{X_{VG}(t)}, \\
X_{VG}(t) &= \mu_{VG} t + \theta\Gamma(t;1,\beta) + \sigma_{VG} W(\Gamma(t;1,\beta)),
\end{aligned}
$$

with: $\mu_{VG} = r + \omega$ where $\omega = \frac{1}{\beta}\log\left(1 - \beta\theta - 0.5\beta\sigma_{VG}^2\right)$.

When simulating $S(t)$, we can directly get all the samples without generating paths, since $W(\Gamma(t;1,\beta)) \stackrel{\mathrm{d}}{=} \sqrt{\Gamma(t;1,\beta)}Z$, where $Z \sim \mathcal{N}(0,1)$ and where $\Gamma(t;1,\beta)$ can be

simulated from the gamma distribution with scale parameter T/β and shape parameter β.
Additionally, in this experiment we sampled via the inverse of the corresponding CDF. The computer code below the Python icon provides us with the plot in Figure 7.

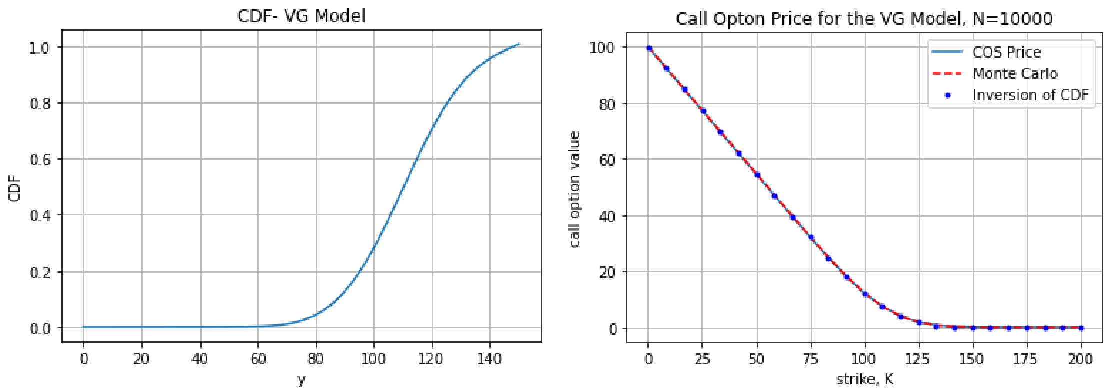

Figure 7: Results for Exercise 9.11 VG. Comparison of the COS and Monte Carlo methods for the Variance Gamma model. Parameters chosen are, $S(t_0) = 100.0$, $r = 0.05$, $\sigma_{VG} = 0.1$, $\beta = 0.2$, $\theta = -0.14$ and $T = 2.0$.

b. For the CGMY model, we follow the derivations from Section 5.4.2. The solution for the CGMY model is given by the following set of equations:

$$\begin{aligned} S(t) &= S(t_0)\mathrm{e}^{X_{CGMY}(t)}, \\ X_{CGMY}(t) &= \left(r + \bar{\omega} - \frac{1}{2}\sigma^2_{CGMY}\right) t + \theta \bar{X}_{CGMY}(t) + \sigma_{CGMY} W(\bar{X}_{CGMY}(t)) \end{aligned}$$

where

$$\begin{aligned} \varphi_{CGMY}(u,t) &= \mathbb{E}\left[\mathrm{e}^{iu\bar{X}_{CGMY}(t)}\right] \\ &= \exp\left(tC\Gamma(-Y)\left((M-iu)^Y - M^Y + (G+iu)^Y - G^Y\right)\right) \end{aligned}$$

and where $\bar{\omega} = -\frac{1}{t}\log(\varphi_{CGMY}(-i,t))\mathrm{i}$.
The computer code below the Python icon provides us with the plot in Figure 8.

c. The NIG case is as the above cases, i.e. once the characteristic function is determined we sample by inversion of the CDF. The computer code below the Python icon provides us with the plot in Figure 9.

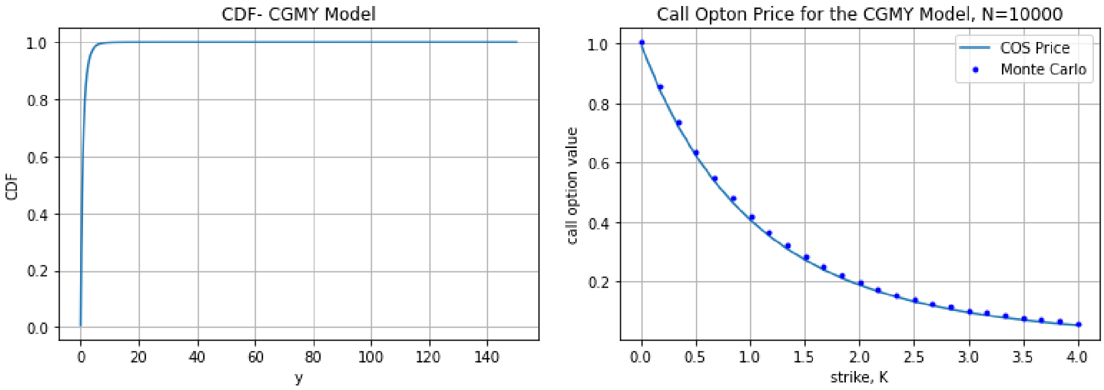

Figure 8: Results for Exercise 9.11 CGMY. Comparison of the COS and Monte Carlo methods for the CGMY model, the Monte Carlo simulation was performed via the inverse of the CDF. Parameters chosen are, $S_0 = 1.0$, $r = 0.0$, $\sigma = 0.1$, $T = 2.0$, $C = 1.0$, $G = 1.0$, $M = 5.0$, $Y = 0.5$.

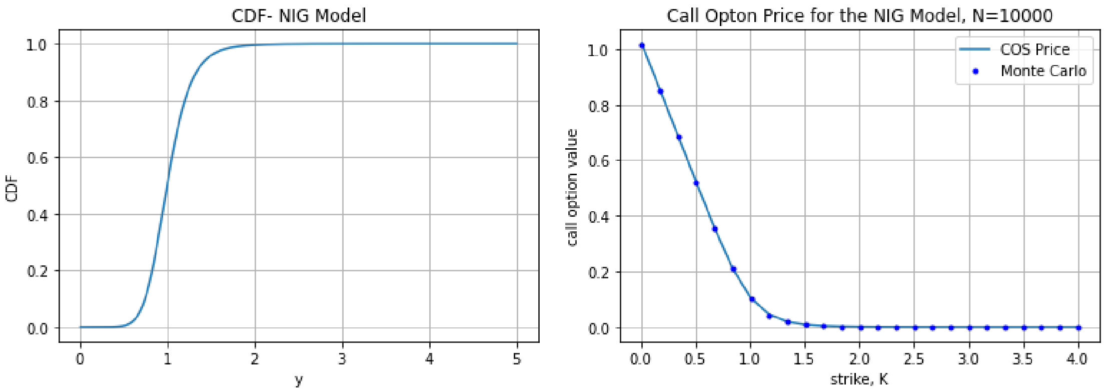

Figure 9: Results for Exercise 9.11 NIG. Comparison of the COS and Monte Carlo methods for the NIG model, the Monte Carlo simulation was performed via the inverse of the CDF. Parameters chosen are, $S_0 = 1.0$, $r = 0.0$, $\sigma = 0.1$, $T = 2.0$, $\delta = 0.1$, $\alpha = 5.5$, $\beta = 0.01$.