Certificate in Quantitative Finance (CQF)

**Session 5.1: Structural Models** *

SOLUTIONS

# 1 Merton (1974): Model Calibration

The solution of this problem involves the following non-linear system

$$E_0 = V_0 N(d_1) - De^{-rT} N(d_2) \tag{1}$$

and

$$\sigma_E E_0 = N(d_1)\sigma_V V_0 \tag{2}$$

For the variables $V_0$ and $\sigma_V$. The first equation comes from the standard expression of the firm's equity as a call option on the value of the assets with a strike price equal to the repayment required by the debt.

The second equation is a bit less straightforward. It comes from the implicit assumption that not only are the firm's assets $V(t)$ modelled as a Geometric Brownian Motion, but also the equity itself. Therefore, we have that for the firm's assets

$$dV_t = rV_t dt + \sigma_V V_t dW_t \tag{3}$$

and for the equity

$$dE_t = rE_t dt + \sigma_E E_t dW_t \tag{4}$$

Applying Ito's lemma to equation (4), it follows that

$$dG = \left(\frac{\partial G}{\partial E} rE + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial E^2}\sigma_E^2 E^2\right) dt + \frac{\partial G}{\partial E}\sigma_E E dW \tag{5}$$

Equating the coefficients which multiply the Brownian motions in both Equations (4) and (5), we obtain

*Tutor: Dr Alonso Peña (alonso.pena@sdabocconi.it )

$$\sigma_E E_t = \frac{\partial G}{\partial V_t} V_t \sigma_V$$

Noting that

$$\frac{\partial E_t}{\partial V_t} = N(d_1)$$

and re-arranging

$$\sigma_V = \frac{E_t}{V_t} \sigma_E N(d_1)$$

### 1.1 Part a

Using the parameters from the statement of the problem and using a numerical method to solve the non-linear system, we obtain:

$V_0 = 12.4572$ and $\sigma_V = 17.83\%$. Once with these values, the probability of default is easily calculated as $N(-d_2) = 7.77\%$

### 1.2 Part b

The solution for various values of equity volatility is:

- $\sigma_E = 10\%$: $V_0 = 12.5123, \sigma_V = 2.39\%$
- $\sigma_E = 20\%$: $V_0 = 12.5123, \sigma_V = 4.79\%$
- $\sigma_E = 30\%$: $V_0 = 12.5123, \sigma_V = 7.19\%$
- $\sigma_E = 40\%$: $V_0 = 12.5116, \sigma_V = 9.61\%$
- $\sigma_E = 50\%$: $V_0 = 12.5068, \sigma_V = 12.11\%$
- $\sigma_E = 60\%$: $V_0 = 12.4914, \sigma_V = 14.82\%$

See for details: Hull JC, Options, Futures, and Other Derivatives, 7th International Edition 2009, page 490.

## 2 Black and Cox (1976): Default Probabilities

This problem challenges the reader to explore the important stochastic calculus tool known as "‘the reflection principle"’ in quantitative finance. A thorough treatment is beyond the scope of the course, but an excellent reference is $www.bus.lsu.edu/academics/finance/faculty/dchance/Instructional/TN00-07.pdf$.

More important than coming up with the exact proof, the reader is invited to learn the conceptual basis of the methodology. In the following we outline some of the key concepts paraphrasing the above reference.

**Brownian Motion**

Suppose the Brownian motion process is the random variable $Z$, which evolves over the time interval from 0 to $T$. We start at $Z(0) = 0$ and ultimately end at $Z(T)$. Assume there is no drift and a volatility of $\sigma$. Consequently, Z(T) has a variance of $\sigma^2 T$. The results for the probability of the maxima and minima of a continuous stochastic process $Y(t)$ can be shown to be:

$$Pr(maxY(t) \geq a) = 2Pr(Y(T) \geq a)$$
$$Pr(minY(t) \leq a) = 2Pr(Y(T) \leq a)$$

**Geometric Brownian Motion**

The probability that a lower barrier $H$ is hit at a future time by a GBM process $dS(t) = \mu S(t)dt + \sigma S(t)dZ$ can be written

$$Prob(S(t) \leq H) \forall t$$

This can be written alternatively as,

$$Prob(minS(t) \leq H).$$

We want to compute $Pr(S(t) \leq H)$ for some $t$. In other words, we want $S(t)$ to fall below $H$ at some time prior to $T$, the option expiry. This can be computed as

$$1 - Prob(S(T) > H, minS(t) > H).$$

This represents the case where $S(t)$ does not go through the barrier and ends up above the barrier, i.e. $S(t)$ not hitting the barrier. Thus, 1 - that probability is the probability of $S(t)$ hitting the barrier. We can then express this as

$$\begin{aligned}
&1 - \Pr(S(T) > H, \min S(t) > H) \\
&= 1 - (\Pr(S(T) > H) - \Pr(S(T) > H, \min S(t) < H)) \\
&= 1 - \Pr(S(T) > H) + \Pr(S(T) > H, \min S(t) < H) \\
&= \Pr(S(T) < H) + \Pr(S(T) > H, \min S(t) < H)
\end{aligned}$$

The first probability is simply:

$$\Pr(S(T) < H) = 1 - N\left(\frac{\ln(S/H) + rT}{\sigma\sqrt{T}}\right)$$

Note the similarity to the Black-Scholes formula.

The second probability is obtained by the drift-adjusted reflection principle:

$$\Pr(S(T) > H, \min S(t) < H) = \left(\frac{H}{S}\right)^{2r/\sigma^2} N\left(\frac{\ln(H/S) + rT}{\sigma\sqrt{T}}\right)$$

where we have used the substitution

$$\left(\frac{H}{S}\right)^{2r/\sigma^2} = \exp\left[\frac{2r\ln(H/S)}{\sigma^2}\right]$$

# 3 Default Re-defined

In this simple exercise, the firm's value process $dV$ is discretized using a nuerical method such as Euler. A Monte Carlo algorithm is then setup such that for each of the simulated paths the Merton and Black & Cox default conditions are checked. The number of paths going in default divided by the total number of simulations gives us an approximation to the probability of default. For the actual calculation the Excel Spreadsheet shown in class could be used. As a complement below you can find a Matlab algorithm and some graphs demonstrating the results. For the parameters given, the estimated default probability computed with the Matlab algorithm shown is 59% (with 10,000 simulations and 100 time-steps).

```
%%% Monte Carlo Simulation Default Probabilities
%%% Default re-defined: Merton + Black & Cox Models
%%% Alonso Pena,   30.4.2009

clear all; close all;

%-----------------------------------------------------------------
% STEP 0: Initialize input data
%-----------------------------------------------------------------

V0 = 100;
ru = 0.05;
sigma=0.40;
K = 90;
D=80;
T = 1;
M=101; % total num timesteps
```

```
nsim = 1000;  % total num mc simulations
default_count=0;
dt=T/M;

%%% MAIN Monte Carlo simulations
for i = 1:nsim

default_flag=false;
VOld=V0;

for j=1:M % time integration

%-----------------------------------------------------------------
% STEP 1: Generate draw from N(0,1), std normal distribution
%-----------------------------------------------------------------
phi = randn;

%-----------------------------------------------------------------
% STEP 2: Integrate SDE for one timestep
%-----------------------------------------------------------------
VNew = VOld + ru*VOld*dt + sigma*VOld*sqrt(dt)*phi;
V(i,j)=VNew;

%-----------------------------------------------------------------
% STEP 3: Check if barrier touched (i.e. default) before maturity
%-----------------------------------------------------------------

%%% BLACK-COX
if (VNew<D)
  default_flag=true;
end

VOld=VNew;

end % time integration

VT(i) =VNew;

%-----------------------------------------------------------------
% STEP 4: Check if barrier touched (i.e. default) at maturity
%-----------------------------------------------------------------
%%% MERTON
if VT(i)<K
    default_flag=true;
```

```
end

%-----------------------------------------------------------------
% STEP 5: Update default flag for each simulation
%-----------------------------------------------------------------
if  default_flag==true;
 default_count=default_count+1;
end

end % simulations

%-----------------------------------------------------------------
% STEP 6: Plot the mean of distribution ST and percent default
% probability
%-----------------------------------------------------------------
meanVT=mean(VT) % mean of distribution
default_count/nsim % percent defaults

%-----------------------------------------------------------------
%%% STEP 7: figures (optional)
%-----------------------------------------------------------------

figure(1)
plot(1:M,V(1:100,:),'b-','LineWidth',1,'Color','blue')
xlabel('TIME: t');
ylabel('FIRM VALUE: V(t)');
axis([0 M 0 300])
grid

figure(2)
hist(VT,30)
xlabel('V(T)'); ylabel('count');
% title(parameters)
grid

figure(3)
x = [default_count nsim-default_count];
% explode = [1 0];
% pie3(x,explode)
pie3(x)
title('percent defaults [red] and no default [green] ');
colormap(prism)

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

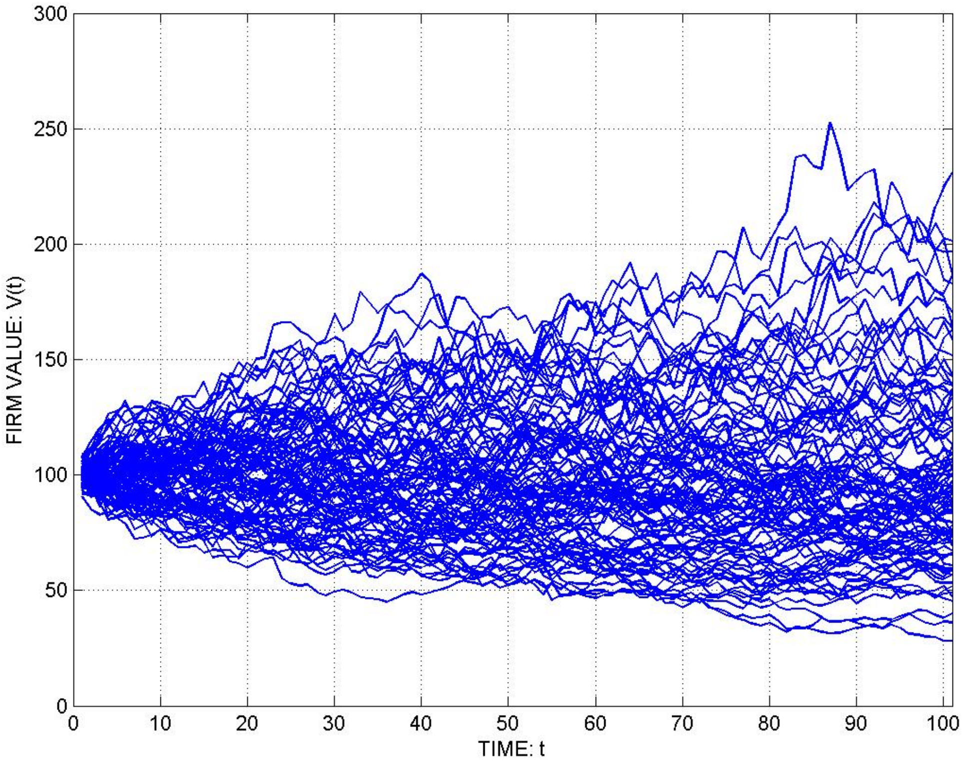


Figure 1: First 100 Monte Carlo paths.

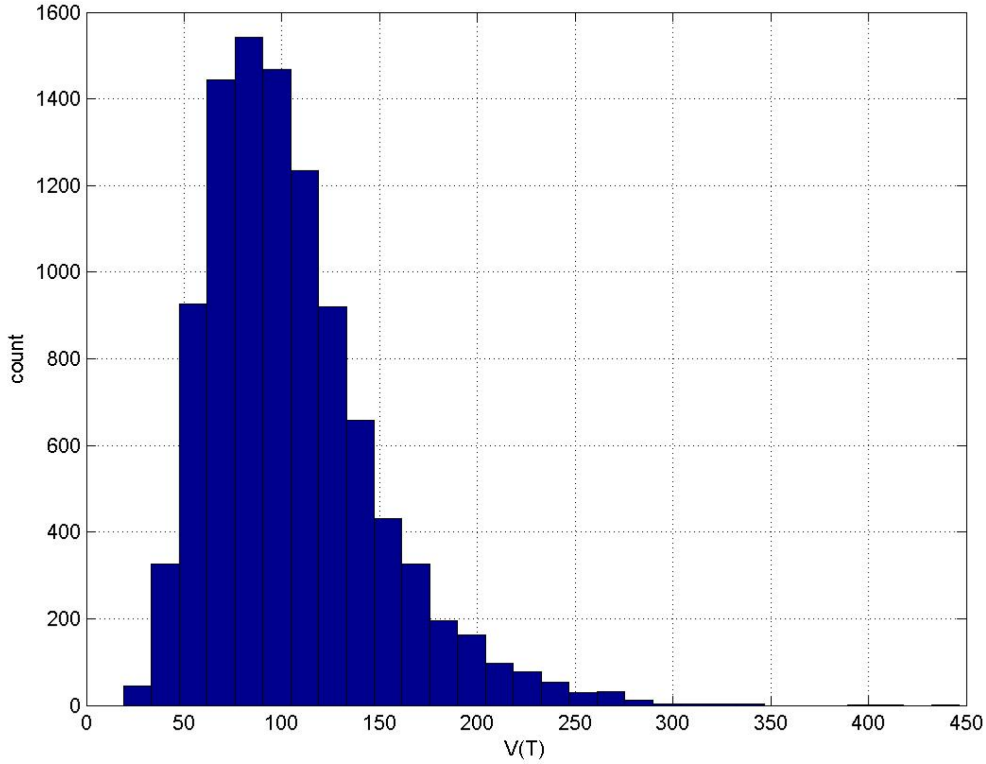


Figure 2: Histogram of $V(T)$.

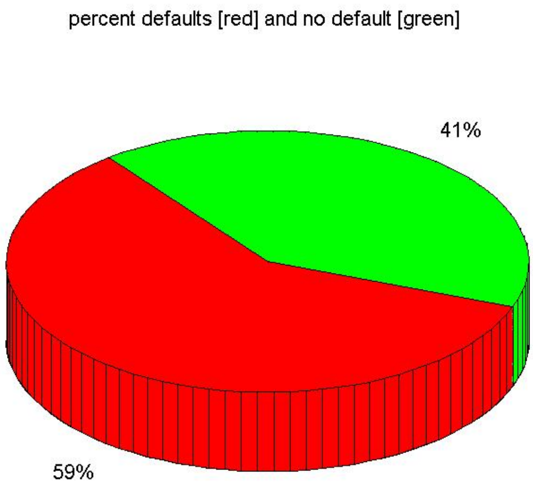


Figure 3: Percent defaults and no defaults.