BEGINNING OF 26/09/2022 RECORDING

14/10/2024

Use of slides about Monte Carlo:
"Monte Carlo Simulations" D. Marazzina.

Example: of a very simple MC simulation.

Pricing (on MATLAB) of a EU call option under B&S assumption.

```
Nsim = 1e6;            discretization parameter
T=1; K=1;              contract parameters
S0 = 1; r = 0.001;     market parameters
sigma = 0.4;           market (CALIBRATED) parameters
```

Indeed, this param. doesn't exist! You have to calibrate it. We'll speak about it.

1) SIMULATE:

```
XT = normrnd( (r-sigma^2/2)*T, sigma*sqrt(T),
                                    Nsim, 1 );
```

normrnd: samples from $\mathcal{N}(\mu, \text{sigma}^2)$

XT: column vector containing all the samples.

ways to sample

We could also do:

```
XT = (r-sigma^2/2)*T + sigma*sqrt(T)*randn(Nsim,1)
```

randn: centered-std-normal

OR:

```
XT = icdf('Normal', rand(Nsim,1), MU, SIGMA )
```

rand: $\mathcal{U}([0,1])$; SIGMA: std

2) Compute the discounted payoff:

```
disc_payoff = exp(-r*T)*max(S0*exp(XT)-K,0);
```

3) Compute the price :

Price = mean(disc_payoff);

4) Get the exact price:

Exact_Price = blsprice(S0,K,r,T,sigma);

Rmk: We just have to change the simulation step of X_T if we want to change the model. The rest stays the same & the MC still works.

What if we don't have an exact price but we want to know the quality of our MC?

$\longrightarrow$ see the slides.

$\longrightarrow$ on MATLAB we have the 'normfit' command. It can also return a 95% Confidence Interval (CI) for the parameter estimates:

For a MC, always prefere not to get the price, but a CI @ a certain level.

MATLAB: [Price,~,CI]= normfit(disc_payoff).

↳ it's better to do this because then, fixing a certain level of confidence, you can be sure on the i^{th} decimal of your estimation. E.g.:

CIlow = 0.158432
CIhigh= 0.158467 } we are sure for the first 3 decimals.

From European to "Path-Dependent":

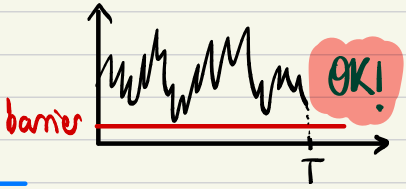
DEF: Path-dependent: the price of the derivative depends not only on the value @ maturity, but also on some values during the life.

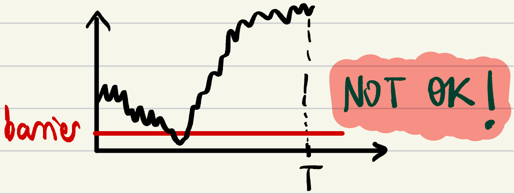
↳ Fortunately, the asset pricing theorem can be extended to path-dependent options.

If you look @ path dependent options, you have two kinds:

1) "Continuous Monitoring":

- down and out barrier: the option can be exercised @ maturity iff the underlying asset price never touched the barrier.





2) "Discrete Monitoring":

- down and out barrier: similar but we only have a set of dates when we check the condition on the barrier. It's OK as long as the underlying asset's price is above the barrier for each date in the set.

You can have different path-dependent options:

- cf Asian Options in MF2;
- arithmetic/geometric average fixed/floating strike ... options.

means the strike can change.

Example: Let's write some code on MATLAB to price some of these particular options.

- cf fct "Asset_BS(so, r, sigma, T, Nsim, M).m".

  ↳ Now we don't only need the maturity price of the underlying asset!

  We need all the prices of the lifetime of the option, i.e: $\forall t \in [0,T]$.

- cf "Put_Asian_Floating_BS.m":

  to price an Arithmetic Average Floating Strike Asian Put Option, under B&S assumptions, with Weekly Monitoring.

  ↳ We need the arithmetic average:

  mean(X,2) % by ROW.

  So the discounted payoff is:

floating strike!

```
disc_payoff = exp(-r*T)*max(mean(S,2)-S(:,end),0);
```

→ `M = round(52*T);`

- cf "Put_Lookback_Maximum_BS.m":

to price a Fixed Strike Lookback Put on the Maximum under B&S Assumptions, with Monthly Monitoring.

↳ `disc_payoff = exp(-r*T)*max(K-max(S,[],2),0);`

Put &:

`max(S,[],2)` : max row by row.

Example: • Let's look at how to price a "Put Down & Out Barrier Option" on MATLAB:

↳ We need a new parameter: the barrier, D.

↳ We need to take the barrier into account:

$$\text{disc\_factor} = \exp(-r*T)*\max(K-S(:,\text{end}),0).*\underbrace{(\min(S,[\,],2)>D)}_{\begin{cases}\text{if it's true}, = 1\\ \text{otherwise}, = 0\end{cases}};$$

↳ D&O Barrier options are then cheaper than classical options: so they are used for hedging.

- Let's price a Floating Strike Lookback Call on the minimum under B&S Assumptions:

↳ `disc_payoff = exp(-r*T)*max(S(:,end)-min(S,[],2),0);`

- Let's price an Up & In Barrier Call under B&S Assumptions:
  - ↳ we need an upper barrier : U.
  - ↳ disc_payoff = exp(-r*T)* max(S(:,end)-K,0).*(max(S,[],2)>U);

- Let's price a Knock & Out (double) Barrier Call under B&S Assumptions:
  - ↳ We need two barriers: D, U (D<U).
  - ↳ disc_payoff = exp(-r*T)* max(S(:,end)-K,0) .*(min(S,[],2)>D) .*(max(S,[],2)<U);

So, basically, we know how to price any kind of derivatives with the exception of American options (which for the moment we don't consider).

END OF 26/09/2022 RECORDING

30/09/2024 Recording

For MATLAB code : see the folder "Lecture 08 – MC".

15/10/2024

Variance Reduction :

Why ? Bc in the classical MC schema, the standard error of a MC $= \frac{std}{\sqrt{N_{sim}}}$.

So : $\exists$ 2 approches to decrease the (standard) error of a MC :

- increase N_{sim} ; ⇝ VERY SLOW TO CONVERGE !
- decrease the variance.

OK if 1 time a year, but not OK if everyday.

X is one RV, & we want to compute $\theta = \mathbb{E}[X]$.

Imagine we have Y, a RV, s.t $\theta = \mathbb{E}[Y]$ also and we know that $Var(X) > Var(Y)$.

Goal : to compute $\theta = ?$

It is better to estimate θ by MC ON Y, because the CI is $\left(\hat{\theta} - \frac{\sigma_Y}{\sqrt{n}} a \, ; \, \hat{\theta} + \frac{\sigma_Y}{\sqrt{n}} b\right)$ which is smaller than the CI with X since $\sigma_X > \sigma_Y$.

First technique : Antithetic Variable :

X RV s.t $\theta = \mathbb{E}[X]$.

Let's take $X_1, X_2 \sim X$ and take $Y = \frac{X_1 + X_2}{2}$.

Then : $\mathbb{E}[Y] = \frac{1}{2}\left(\mathbb{E}[X] + \mathbb{E}[X]\right) = \mathbb{E}[X] = \theta$.

- What about the variance of Y ?

$$Var(Y) = \frac{1}{4}\left(Var(X_1) + Var(X_2) + 2\,cov(X_1, X_2)\right)$$

$$= \frac{1}{2}\left(Var(X) + Cov(X_1, X_2)\right).$$

Different cases:

- $\cancel{X_1 \perp\!\!\!\perp X_2}$: then $Var(Y) = \frac{1}{2} Var(X) < Var(X)$.

But not a good idea since the computational cost to simulate Y is $\sim$ twice the one to simulate X, so it's not worth it.

- $\rho(X_1, X_2) < 0$: NEGATIVE CORRELATION.

How to get this ?

For example, our classical situation is when we want to price a EU call under B/S :

$$X = e^{-rT} \max\left(S_0 e^{(r - \frac{\sigma^2}{2})T + \sigma\sqrt{T} Z} - K ; 0\right)$$

source of randomness.

So we have : $X = g(Z)$, where

g is (weakly) monotone. The idea is:

if I take $\left.\begin{matrix} z_1 \\ z_2 \end{matrix}\right\} \sim N(0,1)$ & negatively correlated, then $g(z_1)$ & $g(z_2)$ are still negatively correlated (because of monotonicity of g).

We would like: perfectly negative correlated. So it's easy:

- sample $z \sim N(0,1)$;
- $z_1 = z$ & $z_2 = -z$;

$\Big\}$ Same computational cost as classical MC.

In practice: $Y = e^{-rT} \max(S_0 e^{rT+X} - K; 0)$

with X Lévy or "Stochastic Volatility" proc.

If we want to sample, in MATLAB we use

Usual situation

two sources of randomness:

$$\text{SAMPLE} \begin{cases} \text{randn} & \begin{cases} x_1 = \text{randn} \\ x_2 = -x_1 \end{cases} \quad [\text{Normal}] \\[2ex] \text{rand} & \begin{cases} x_1 = \text{rand} \\ x_2 = 1 - x_1 \end{cases} \quad [\text{uniform}] \end{cases}$$