Monte Carlo Simulation for Pricing European and American Basket option

Giuseppe Bruno
Bank of Italy

The R User Conference 2010, Gaithersburg, Maryland
July 20-23

OUTLINE

1. Motivation and focus of the paper
2. Mathematical Framework
3. The Monte Carlo Simulator
4. Computing the Greeks of the Options
5. Implementing the Longstaff-Schwarz method for American Options
6. Some numerical Examples
7. Concluding remarks

1 . Motivation and focus of the paper

Market economies hinge heavily on the use of derivative securities for achieving an optimal risk management.

Firms are confronted with different financial risks and look for the most efficient way to edge their risks.

Employment of basket options is a very efficient avenue for achieving the wished hedging position.

i. No closed form solution is available for pricing basket of options;
ii. Quite often it is assumed that Monte Carlo techniques are not suitable for multivariate option pricing problems.

2. Mathematical framework

Pricing an option boils down to the computation of the following discounted expectation:

$$P_t = e^{\int_t^T (-r(T-t)d\tau)} E_Q[g(T,S)]$$

where:

$g(T,S)$ is the option's payoff at expiration when the underlying stock has value S;

E_Q is the expectation operator w.r.t. expiration stock value distribution.

2. Mathematical framework

In the paper we focus on pricing an option on a portfolio of the following kind:

$$V_t = \sum_{i=1}^{n} a_i \cdot S_i$$

each stock S_i follow a geometric brownian motion. The pool of stocks shows a given correlation structure.

$$\begin{cases} \dfrac{dS_1}{S_1} = r\,dt + \sigma_1\,dw_1 \\ \dfrac{dS_2}{S_2} = r\,dt + \sigma_2\,dw_2 \\ \vdots \\ \dfrac{dS_n}{S_n} = r\,dt + \sigma_n\,dw_n \end{cases}$$

2. Mathematical framework

The system of Brownian motions is transformed into a recursive system by employing the Cholesky decomposition of the correlation matrix.

A set of *M* sample paths is generated for each stock.

Now it is possible to compute the option's payoff for each path of the sample: $g^s(T,V)$

The option price and its standard deviation are computed according to:

$$\hat{P} = \frac{1}{M} e^{-r(T-t)} \sum_{s=1}^{M} g^s(T,V)$$

$$\sigma_{\hat{P}} = \sqrt{\frac{1}{M}\left(\sum_{s=1}^{M}\left(P_s - \hat{P}\right)^2\right)}$$

3. The Monte Carlo Simulator

The simulator is based on a set of R functions:

1) Generation of the gaussian innovations for all the Brownian paths (RNG),
2) Generation of the replications for every stock composing the portfolio,
3) Computation of the portfolio value,
4) Evaluation of the elements for path dependent options,
5) Computation of the basket option value

3. The Monte Carlo Simulator

As example we consider the model required for an arithmetic Asian option:

```
MODEL
FUNCTION> DIVBYT 1
PARAMETER> RFREE, SIGMA, DT, TMATU, STRIKE
COMMENT> Stock Price dynamics of the asset
IDENTITY> Stock1
EQ>Stock1 = Lag(Stock1)*EXP((RFREE-SIGMA**2/2)*DT+(SIGMA*EPSIL1)*SQRT(DT))
COMMENT> Partial sum of the portfolio
IDENTITY> PARPORT
EQ> PARPORT = Lag(PARPORT) + Stock1
COMMENT> arithmetic average
IDENTITY> AVEPORT
EQ> AVEPORT = DIVBYT(PARPORT)
COMMENT> pricing the Asian call on the arithmetic average.
IDENTITY> CASIAF
EQ> CASIAF = EXP(-RFREE*TMATU)*max(0.0,AVEPORT-STRIKE)
```

3. The Monte Carlo Simulator

The variable EPSIL1 is used to feed in the stochastic disturbances.

The previous MODEL is initially translated into an executable Fortran module,

in a second phase the MODEL is simultaneously solved for all the required replications,

means and standard deviations are automatically generated, higher order statistics can be computed by the user.

3. The Monte Carlo Simulator

For the goal of reducing the sampling variance of the replications the simulation engine provides three techniques:

1) antithetic variates
2) Quasi Monte Carlo methods (Low Discrepancy Sequences)
3) Control Variates

The first two methods are implemented by generating particular values for the stochastic disturbances.

Implementation of the Control Variates methods requires the modification of the MODEL equations.

Basic code example

```
## begin simulation
for(j in 1:trials){
  for(i in 2:(m+1)){
    z1 <- rnorm(1,0,1)
    z2 <- rnorm(1,0,1)
    z3 <- rnorm(1,0,1)

    ds1 <- s1[i-1]*(r[i-1]*dt + s1.vol*sqrt(dt)*z1)
    ds2 <- s2[i-1]*(r[i-1]*dt + s2.vol*sqrt(dt)*(rho*z1 + sqrt(1-rho^2)*z2))
    dr  <- k*(theta - r[i-1])*dt + beta*sqrt(dt)*z3

    s1[i] <- s1[i-1] +  ds1
    s2[i] <- s2[i-1] +  ds2
    r[i]  <- r[i-1]  +  dr
  }
  ss <- sum(r[2:(m+1)]*dt)
  c[j] <- ifelse(s1[m+1]>K1 && s2[m+1]>K2, exp(-ss), 0)
}

cat("Option Price Estimate:", round(mean(c),3), "\n")
cat("Standard Error:", round(sd(c)/sqrt(trials),3), "\n")
```

Basic code example

The previous Monte Carlo simulation code is based on a double loop:

1) the innermost loop generates one simulation path for two stocks and the riskless interest rate

2) the outermost loop runs over the trials numbers generating different random paths

Option price and its standard error is computed at the end by averaging the option values of each replication.

Here there is no saving for the different simulation paths

A different option requires the insertion of code into the outermost loop

Basic code example

```
## begin simulation
z1 <- matrix(rnorm(trials*m,mean=0,sd=1), trials,m)
z2 <- matrix(rnorm(trials*m,mean=0,sd=1), trials,m)
z3 <- matrix(rnorm(trials*m,mean=0,sd=1), trials,m)

  for (i in 2:(m+1)){
    ds1[,i-1] <- s1[,i-1]*(r[,i-1]*dt+s1.vol*sqrt(dt)*z1[,i-1])
    ds2[,i-1] <- s2[,i-1]*(r[,i-1]*dt+s2.vol*sqrt(dt)*(rho*z1[,i-1]+sqrt(1-rho^2)*z2[,i-1]))
    dr[,i-1]  <- k*(theta- r[,i-1])*dt + beta*sqrt(dt)*z3[,i-1]
    s1[,i] <- s1[,i-1] +  ds1[,i-1]
    s2[,i] <- s2[,i-1] +  ds2[,i-1]
    r [,i] <- r [,i-1] +  dr [,i-1]
}
  ss <- rowSums(r[,seq(2,(m+1))]*dt)
  c <- ifelse(s1[,m+1]>K1 & s2[,m+1]>K2, exp(-ss), 0)
cat("Option Price Estimate:", round(mean(c),10), "\n")
cat("Standard Error:", round(sd(c)/sqrt(trials),10), "\n")
```

Basic code example

This version of the Monte Carlo simulation code is based on a single loop (time) and the use of two dimensional matrices:

the loop generates all simulation paths for the two stocks and the riskless interest rate at each time period.

The results are stored in matrices with rows referring to replications and the columns referring to time periods.

Option price and its standard error is computed at the end by averaging the option values of each replication.

All the different simulation paths are available for other statistics.

Different options price can be evaluated outside of the loop

Performance Comparison

AMD OPTERON 4 proc 4 core 2.7 GHz X86_64 platform with RHEL 5.4 RAM 32 GB

SINGLE LOOP EXECUTION

Time range subdivided in 200 intervals.
10^4 Monte Carlo trials takes 3 seconds
10^5 Monte Carlo trials takes 26 seconds
10^6 Monte Carlo trials takes 4’ 42 seconds

DOUBLE LOOP EXECUTION

Time range subdivided in 200 intervals.
10^4 Monte Carlo trials takes 1’ 37 seconds
10^5 Monte Carlo trials takes 15’ 54 seconds
10^6 Monte Carlo trials takes 3^h 9’ 38 seconds

4. Computing the Greeks of the Options

Computing derivatives approximations is always a tricky business

In our Monte Carlo simulation software the following three methods for the estimation of the greeks have been implemented and tested:

1) resimulation
2) the pathwise method
3) the likelihood ratio method

All these methods requires some modifications in the MODEL for providing the option price sensitivities.

5. Implementing the Longstaff-Schwarz method for American Options

Efficient pricing of American option is still a thorny issue in both algorithmic complexity and computational burden.

American option can be exercised any time prior expiration, therefore these derivatives embeds an optimal stopping time problem.

Given the availability of our Monte Carlo simulation engine, the implementation of the Least Squares Monte Carlo (LSM) method, proposed by Longstaff and Schwarz (2001), is seemed a simple solution.

5. Implementing the Longstaff-Schwarz method for American Options

At each simulation time we compare the immediate exercise value with the expectation of the continuation holding policy.

The value of continuing the option life at time t_j is given by:

$$F(t_j) = E_Q\left[\sum_{k=j+1}^{K} e^{-\left(\int_{t_j}^{t_k} r(\omega,\tau)d\tau\right)} C(\omega, t_k; t_j, T)\right]$$

$C(\omega, t_k; t_j, T)$ is the cash flow conditional on the option existence

at time t_k and optimal behaviour up to t_j

5. Implementing the Longstaff-Schwarz method for American Options

The expected value from continuing the option's life is computed by regressing the realized cash flows on a set of orthogonal basis functions.

In our examples we have employed the Laguerre polynomials:

$$L_n(x) = \frac{e^x}{n!} \cdot \frac{d^n}{dx^n}\left(x^n e^{-x}\right)$$

we have used only the first three polynomials:

$$L_0(x) = 1$$
$$L_1(x) = 1 - x$$
$$L_2(x) = 1 - 2x + x^2/2$$

5. Implementing the Longstaff-Schwarz method for American Options

Different orthogonal polynomials have been proposed in the literature (Hermite, Legendre, Chebyshev)

The number of polynomials and their coefficients can be easily adjusted by the user.

6. Some numerical examples

European basket option: price of a call and a put on a weighted average of 3, 5 and 10 assets.

Table 1: European basket option: price of a call and a put option on a weighted average of 3, 5 and 10 assets

n. assets	S_0/K	Call M.C.	Call An. Va.	Put M.C.	Put An. Va.
3	1	43.914	43.652	2.977	3.226
		(.384)	(.384)	(.095)	(.095)
3	.9	83.018	82.645	0.151	0.178
		(.424)	(.426)	(.017)	(.019)
3	1.1	16.452	16.395	17.667	18.011
		(.265)	(.264)	(.233)	(.3)
5	1	44.029	43.740	2.752	2.907
		(.377)	(.376)	(.086)	(.088)
5	.9	83.432	82.995	0.114	0.12
		(.414)	(.415)	(.014)	(.015)
5	1.1	16.183	15.997	16.947	17.204
		(.261)	(.258)	(.225)	(.229)
10	1	43.837	43.722	2.905	2.977
		(.379)	(.381)	(.088)	(.089)
10	.9	83.086	82.915	.113	.129
		(.418)	(.420)	(.015)	(.015)
10	1.1	16.165	16.174	17.274	17.469
		(.263)	(.263)	(.228)	(.23)

Note: below the price standard errors in parentheses.
Monte Carlo with 10,000 replications.

6. Some numerical examples

Asian basket option: price of a call on a weighted average of 3, 5 and 10 assets.

n. assets	S_0/K	Arith ave	Geom ave	Lookback	Chooser
3	1	54.051 (.394)	52.168 (.378)	112.902 (.772)	111.712 (.751)
3	.9	94.775 (.415)	92.852 (.399)	153.91 (.789)	151.46 (.783)
3	1.1	21.991 (.299)	20.389 (.280)	75.604 (.714)	79.766 (.660)
5	1	54.433 (.387)	52.565 (.372)	113.882 (.757)	112.522 (.739)
5	.9	95.325 (.406)	93.425 (.391)	155.03 (.772)	152.63 (.767)
5	1.1	22.013 (.294)	20.414 (.276)	76.074 (.704)	79.506 (.655)
10	1	54.071 (.392)	52.213 (.376)	113.20 (.764)	111.88 (.746)
10	.9	94.964 (.410)	93.073 (.395)	154.32 (.779)	151.92 (.774)
10	1.1	21.949 (.297)	20.368 (.279)	75.621 (.710)	79.313 (.658)

Note: below the price there is the standard errors in parentheses.

6. Some numerical examples

American option: price of a put on a single asset and a basket of 3 assets (preliminary results).

n. assets	S_0/K	put M.C.	call M.C.
3	1	2.359 (.089)	14.482 (.143)
3	.98	1.218 (.064)	8.254 (.122)
3	1.03	5.140 (.013)	27.517 (.147)
5	1	2.125 (.083)	13.729 (.14)
5	.98	1.012 (.058)	7.613 (.11)
5	1.03	4.741 (.129)	26.757 (.14)
10	1	2.282 (.085)	13.761 (.136)
10	.98	1.086 (.06)	7.612 (.115)
10	1.03	4.968 (.013)	26.789 (.14)

Note: below the price there is the standard errors in parentheses.

7. Concluding remarks

We have shown a software tool for employing Monte Carlo simulation methods for pricing European and American basket options and many kind of path dependent options.
A very critical role is played by the Random Number Generator (found relevant differences with MATLAB)
The software tool is based on a set of R functions.
It allows end users to mathematically describe their multi asset portfolio.
The same technical apparatus has been adopted for the computation of the greeks of the options.

Thank you for your attention.

Giuseppe Bruno
Bank of Italy
Research and International Relations
Head of I.T. Support Unit
giuseppe.bruno@bancaditalia.it