Appendix : Super Quant Monte Carlo Challenge 2008

Organized by ERCIM, ETSI and INRIA
Sophia Antipolis, France,
Monday 20 - Friday 24 October 2008.

September 17, 2008

1 Introduction

This document provides mathematical representation of the Super Quant Challenge. We assume that the reader has acquired basic knowledge of Derivative/Option pricing, Black/Scholes model and Monte Carlo simulations for option pricing. The references provided in the problem statement can be useful to create sufficient background to facilitate the understanding of the concepts described here. The java implementations of the pseudocodes described in this document are available on the website. The participants can refer to the source code and this appendix simultaneously for better understanding.

2 Black Scholes model defined for d-dimensional options

The well known Black-Scholes model describes the evolution of a stock price through the stochastic differential equation,

$$\frac{dS(t)}{S(t)} = (r - D)dt + \sigma dW(t) \tag{1}$$

where,

1. r is the interest risk free rate.
2. D is the dividend.
3. W a standard Brownian motion.

This equation models the percentage change $\frac{dS}{S}$ in the stock price as increments of a Brownian motion. The solution of (1) is,

$$S(T) = S(0)\exp([(r - D) - \frac{1}{2}\sigma^2]T + \sigma\sqrt{T}Z) \tag{2}$$

where Z is a standard normal random variable (with mean 0 and variance 1).

Extending this model for a d dimensional option, alternatively termed as a basket of d assets, the asset prices for d underlying assets can be modeled as follows,

$$dS_t^i = (r - D)S_t^i dt - \sigma S_t^i L\, dB_t^i, \tag{3}$$

where,

1. S_t^i are asset prices at time t, for $i = 1, .., d$. S_0^i, at time 0.
2. σ is a volatility of the asset prices.
3. L is a lower-triangular matrix derived from Cholesky decomposition of a given correlation matrix between the assets in the basket.
4. $B_t = (B_t^1, .., B_t^d)$ is an independent Brownian motions vector.

Assuming C as a (positive definite symmetric) correlation matrix of d assets, L is computed such that $C = LL^T$. Given C, L can be calculated by the Cholesky decomposition. We choose $C := (\rho_{i,j}$ of the form $\rho_{i,i} = 1$, $\rho_{i,j} = \rho$, $i \neq j$ with $-1 < \rho < 1$.

3 Monte Carlo methods

To illustrate Monte Carlo methods, we consider calculation of the option price which is the expected present value of the payoff of an option. Let us take an example of a vanilla average call option. The payoff of such call option at the maturity date T is given as,

$$\Phi(S_t^i, t) = (\sum_{i=0}^{d} w_i S_T^i - K)^+ \tag{4}$$

where K is the strike price, w_i is the weight of the ith asset in the basket (i.e. $w_i = \frac{1}{d}$) and $i = 1, .., d$. We denote the expected present by $\mathbb{E}[e^{-rT}\Phi(S_T^i, T)]$. The Algorithm 1 gives a pseudo–code that illustrates the steps in simulating a number of Monte Carlo paths ($nbMC$) by using a discrete (Euler) approximation. We use Z_{ikj} to denote the kth draw from the normal distribution along the ith path of the jth asset. In this setting, we partition $[0, T]$ interval into N_T subintervals t_k of length $\Delta t = \frac{T}{N_T} = t_k - t_{k-1}$, with $k = 0, .., N_T$. The Algorithm 1 can be applied for pricing high–dimensional options which do not have any analytical solutions. In the particular case of a vanilla basket option, the mechanism for generating paths can be simplified as in the Algorithm 5.

4 Greeks hedging

Let us define Delta Δ, Gamma Γ, Speed, Rho ρ and Theta θ definitions as follows:

1. The Δ of an instrument is the mathematical derivative of the option value P with respect to the underlying price, $\Delta = \frac{\partial P}{\partial S}$.
2. The Γ is the second derivative of the value function with respect to the underlying price, $\Gamma = \frac{\partial^2 P}{\partial S^2}$
3. The Speed is the third derivative of the value function with respect to the underlying price, Speed = $\frac{\partial^2 P}{\partial S^3}$
4. The ρ is the first derivative of the value function with respect to the interest free rate, $\rho = \frac{\partial P}{\partial r}$
5. The Θ is the first derivative of the value function with respect to the time, $\Theta = \frac{\partial P}{\partial T}$

One of the popular approaches to compute such derivative in computer simulation is finite difference methods. A finite difference is a mathematical expression of the form $f(x + b)f(x + a)$. If a finite difference is divided by $b - a$, one gets a difference quotient. The approximation of derivatives by finite differences plays a central role in finite difference methods for the numerical solution of differential equations, especially boundary value problems. Hence, we have the derivative of a function f at a point x is defined by the limit,

$$\frac{\partial f}{\partial x} = f'(x) = \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon) - f(x)}{\varepsilon}. \tag{5}$$

By appling (5) in the Δ, Γ, Speed, ρ, and Θ hedging we have,

$$\Delta = \frac{\partial P}{\partial S} = \lim_{\varepsilon_S \to 0} \frac{P(S + \varepsilon_S) - P(S)}{\varepsilon_S}. \tag{6}$$

$$\Gamma = \frac{\partial^2 P}{\partial S^2} = \lim_{\varepsilon_S \to 0} \frac{P(S + \varepsilon_S) - 2P(S) + P(S - \varepsilon_S)}{\varepsilon_S^2}. \tag{7}$$

$$\text{Speed} = \frac{\partial^3 P}{\partial S^3} = \lim_{\varepsilon_S \to 0} \frac{P(S + 2\varepsilon_S) - 3P(S + \varepsilon_S) + 3P(S) - P(S - \varepsilon_S)}{\varepsilon_S^3}. \tag{8}$$

$$\rho = \frac{\partial P}{\partial r} = \lim_{\varepsilon_R \to 0} \frac{P(r + \varepsilon_R) - P(r)}{\varepsilon_R}. \tag{9}$$

$$\Theta = \frac{\partial P}{\partial T} = \lim_{\varepsilon_T \to 0} \frac{P(T + \varepsilon_T) - P(T)}{\varepsilon_T}. \tag{10}$$

5 Sequential pseudo-code for pricing and hedging a vanilla average basket option

The Algorithms 3, 4 provide pseudo-codes for pricing and hedging a vanilla average basket of d assets. These algorithms illustrate how to compute the price of a vanilla put option with average payoff and the Greeks values such as the Delta Δ, Gamma Γ, Speed, Rho ρ and Theta θ. Note that the participants will have to modify the Algorithm 3 for pricing Barrier options (Up-In, Up-Out, Down-In and Down-out). The implementation of the generalized algorithm for pricing call/put vanilla and barrier options is available in the application provided on the plugtest website.

Algorithm 1 Paths simulating of a generic basket of d assets

Require: S_0^j, r, D, σ, N_T, number of simulations $nbMC$

for $i = 0$ to $nbMC$ **do**
    for $j = 0$ to d **do**
        for $k = 0$ to N_T **do**
            $S_{t_k}^j = S_{t_{k-1}}^j \exp(((r - D) - \sigma^2/2)(t_k - t_{k-1}) + \sigma\sqrt{t_k - t_{k-1}} Z_{ikj})$
        end for
    end for
end for

Algorithm 2 Paths simulating of a vanilla basket of d assets

Require: S_0^j, r, D, σ, N_T, number of simulations $nbMC$

for $i = 0$ to $nbMC$ **do**
    for $j = 0$ to d **do**
        $S_T^j = S_0^j \exp(((r - D) - \sigma^2/2)(T) + \sigma\sqrt{T} Z_{ij})$
    end for
end for

Algorithm 3 Pricing Vanilla average basket put option

Require: S_0^j, r, D, σ, T, w, ε_S, ε_R, ε_T, K, correlation matrix C, number of simulations $nbMC$
Ensure: **Price**^{0}, **Variance**, **Lower interval**, **Upper interval**

Get the lowertriangular matrix L using Cholesky decomposition of C.
for $i = 0$ to $nbMC$ **do**
  Vector B of random vector $\in N(0,1)$ then get vector $Z = L \times B$.
  for $j = 0$ to d **do**
    $S_T^{0,j} = S_0^j \exp(((r-D) - \sigma^2/2)T + \sigma\sqrt{T}Z_j)$
    $S_T^{1,j} = S_0^j (1+\varepsilon_S) \exp(((r-D) - \sigma^2/2)T + \sigma\sqrt{T}Z_j)$
    $S_T^{2,j} = S_0^j (1-\varepsilon_S) \exp(((r-D) - \sigma^2/2)T + \sigma\sqrt{T}Z_j)$
    $S_T^{3,j} = S_0^j \exp((((r(1+\varepsilon_R)) - D) - \sigma^2/2)T + \sigma\sqrt{T}Z_j)$
    $S_T^{4,j} = S_0^j \exp(((r-D) - \sigma^2/2)(T(1+\varepsilon_T)) + \sigma\sqrt{(T(1+\varepsilon_T))}Z_j)$
    $S_T^{5,j} = S_0^j (1+2\varepsilon_S) \exp(((r-D) - \sigma^2/2)T + \sigma\sqrt{T}Z_j)$
  end for
  $P_i^0 = \sum_{j=0}^{d} w_j S_T^{0,j}$; $P_i^3 = \sum_{j=0}^{d} w_j S_T^{3,j}$; $P_i^4 = \sum_{j=0}^{d} w_j S_T^{4,j}$
  for $j = 0$ to d **do**
    $P_i^{1,j} = \sum_{l=0, l\neq j}^{d} w_l S_T^{0,l} + w_j S_T^{1,j}$; $P_i^{2,j} = \sum_{l=0, l\neq j}^{d} w_l S_T^{0,l} + w_j S_T^{2,j}$
    $P_i^{5,j} = \sum_{l=0, l\neq j}^{d} w_l S_T^{0,l} + w_j S_T^{5,j}$
  end for
  $X_i^0 = (K - P_i^0, 0)^+$; $X_i^3 = (K - P_i^3)^+$; $X_i^4 = (K - P_i^4)^+$
  for $j = 0$ to d **do**
    $X_i^{1,j} = (K - P_i^{1,j})^+$; $X_i^{2,j} = (K - P_i^{2,j})^+$; $X_i^{5,j} = (K - P_i^{5,j})^+$
  end for
  $\text{Payoff}^0 = \sum_{i=0}^{nbMC} X_i^0$; $\text{PayoffSquare}^0 = \sum_{i=0}^{nbMC} (X_i^0)^2$
  $\text{Payoff}^3 = \sum_{i=0}^{nbMC} X_i^3$; $\text{Payoff}^4 = \sum_{i=0}^{nbMC} X_i^4$
  for $j = 0$ to d **do**
    $\text{Payoff}^{1,j} = \sum_{i=0}^{nbMC} X_i^{1,j}$; $\text{Payoff}^{2,j} = \sum_{i=0}^{nbMC} X_i^{2,j}$; $\text{Payoff}^{5,j} = \sum_{i=0}^{nbMC} X_i^{5,j}$
  end for
end for
$\mathbf{Price}^0 = \frac{\exp(-rT)}{nbMC}\text{Payoff}^0$; $\mathbf{Variance}^0 = \frac{\exp(-rT)}{nbMC}\text{PayoffSquare}^0 - (\text{Payoff}^0)^2$
$\mathbf{Price}^3 = \frac{\exp(-(r(1+\varepsilon_R))T)}{nbMC}\text{Payoff}^3$
$\mathbf{Price}^4 = \frac{\exp(-r(T(1+\varepsilon_T)))}{nbMC}\text{Payoff}^4$
for $j = 0$ to d **do**
  $\mathbf{Price}^{1,j} = \frac{\exp(-rT)}{nbMC}\text{Payoff}^{1,j}$; $\mathbf{Price}^{2,j} = \frac{\exp(-rT)}{nbMC}\text{Payoff}^{2,j}$
  $\mathbf{Price}^{5,j} = \frac{\exp(-rT)}{nbMC}\text{Payoff}^{5,j}$;
end for
// Confidence interval at 95% of the put premium
$\mathbf{Lower\ interval} = \mathbf{Price}^0 - 1.96 * \frac{\sqrt{\mathbf{Variance}}}{\sqrt{nbMC}}$
$\mathbf{Higher\ interval} = \mathbf{Price}^0 + 1.96 * \frac{\sqrt{\mathbf{Variance}}}{\sqrt{nbMC}}$

Algorithm 4 Delta, Gamma, Rho, Theta and Speed hedging European basket put option

Require: $\mathbf{Price}^0$,$\mathbf{Price}^1$,$\mathbf{Price}^2$,$\mathbf{Price}^3$,$\mathbf{Price}^4$,$\mathbf{Price}^5$

Ensure: **Delta**, **Gamma**, **Rho**, **Theta**, **Speed**

for $j = 0$ to d **do**

$\quad \mathbf{Delta}^j = \dfrac{\mathbf{Price}^{1,j} - \mathbf{Price}^0}{S_0^j \varepsilon_S}$ {from Equation (6)}

$\quad \mathbf{Gamma}^j = \dfrac{\mathbf{Price}^{1,j} - 2\mathbf{Price}^0 + \mathbf{Price}^{2,j}}{(S_0^j \varepsilon_S)^2}$ {from Equation (7)}

$\quad \mathbf{Speed}^j = \dfrac{\mathbf{Price}^{5,j} - 3\mathbf{Price}^{1,j} + 3\mathbf{Price}^0 - \mathbf{Price}^{2,j}}{(S_0^j \varepsilon_S)^3}$ {from Equation (8)}

end for

$\mathbf{Rho} = \dfrac{\mathbf{Price}^3 - \mathbf{Price}^0}{r\varepsilon_R}$ {from Equation (9)}

$\mathbf{Theta} = \dfrac{\mathbf{Price}^4 - \mathbf{Price}^0}{T\varepsilon_T}$ {from Equation (10)}