Monte Carlo Simulation

October 6, 2014

In this homework we are going to build Monte Carlo based simulation for pricing options. In particular, you will need to price the following two options

1. The current market conditions are: the price of IBM is 152.35$, the volatility σ is 0.01 per day, and $r = 0.0001$ per day. What is the price of a call option that ends 252 days into the future and the strike price is 165$.
2. The current market conditions are: the price of IBM is 152.35$, the volatility σ is 0.01 per day, and $r = 0.0001$ per day. What is the price of an asian call option that ends 252 days into the future and the strike price is 164$. Note an asian call option will pay the maximum between zero and the average price during the 252 days minus the strike price.

This calculations needs to be done where the stopping criteria is with probability 96% the estimation error is less than 1%.
In this homework you will need to use

1. JODA available at http://joda-time.sourceforge.net/ for the DateTime object
2. Commons math at http://commons.apache.org/proper/commons-math// for calculating the distributions.

The framework for the simulation is going to be as follows:

```
// The interface for creating StockPath. The returned list should be ordered by date
public interface StockPath{
  public List<DateTime,double> getPrices();
}
// The interface for calculating the payout function
public interface PayOut{
   public double getPayout(StockPath path);
}
// The interface for Random vector generator
public interface RandomVectorGenerator{
  public double[] getVector();
}
```

The steps that you will need to complete

1. Implement a random vector generator that generate normally distributed numbers.
2. Implement an Anti-Thetic decorator.
3. Implement a geometric brownian motion stock prices.
4. Implement the payouts for the options discussed above.
5. Test your code. Make few stock paths and calculate the payout.
6. Implement a class that tracks $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $\hat{x}^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$. Use that to calculate $\hat{\sigma}^2 = \hat{x}^2 - \hat{\mu}^2$
7. Implement a simulation manager which assemble the right payout, random vector generator, stock path generator and run the simulation.