WILLIAM HART

INTERESTING NOTES

WILLIAM HART

PUBLISHED BY WILLIAM HART

HTTPS://GITHUB.COM/WILL-HART/INTERESTING-NOTES

Generated September 2015

Contents

List of Figures

List of Tables

We have advanced to new and surprising levels of bafflement

Miles Vorkosigan, Komarr

Introduction

This is a collection of notes on topics that have periodically interested me. The notes are available from github at https://github.com/will-hart/interesting-notes under an Apache 2.0 license.

The book uses the `tufte-book` and `tufte-handout` document classes which are also under the Apache 2.0 License. Feel free to read, suggest topics or submit corrections.

Quantitative Finance

This section deals with some applications of quantitative finance, both computational and theoretical.

Equity Mathematical Models

Equities and finance are heavily influenced by random events (see [1] for a detailed philosophical discussion). As such, the analysis of financial instruments lends itself to a stochastic process.

[1] Nicholas Nasim Taleb. *Fooled By Randomness*. Penguin, 2005

A *Geometric Brownian Motion* (GBM) equation (a form of stochastic differential equation, or SDE) can be used to approximate a random process over time. In its continuous form this looks like [2]:

[2] A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014

$$ds = \mu S dt + \sigma S dW \tag{1}$$

This breaks the movement of a stock price down into two key effects:

- a deterministic effect (the left of the plus sign)
- a stochastic effect (the right of the plus sign)

In the equation, μ is known as *drift*, σ is *volatility*, S is the stock price, dt is the change in time and dW is an increment in a *Weiner process*.

As equity markets are a discrete process, this equation must be transformed into a discrete equation.

$$S_{t+1} = S_t(1 + r\Delta t + \sigma \varepsilon_t \sqrt{\Delta t}) \tag{2}$$

Here ε is a sample from a gaussian distribution with zero mean and standard deviation of 1 (i.e. $N(0, 1)$) and r is the risk free rate of return. This equation can be solved iteratively for a given time period if r, σ, ε and S_0 are provided.

Similar formulations can be determined for foreign exchange:

$$X_{t+1} = X_t(1 + (r_d - r_f)\Delta t + \sigma \varepsilon_t \sqrt{\Delta t}) \tag{3}$$

where r_d and r_f are the domestic and foreign fisk free rates of return.

Structural and Intensity Models

TODO! See [3] chapter 2.

[3] A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014

Monte Carlo Simulation

Monte Carlo simulation is a method of estimating a probabilistic outcome through a high quantity of simulations. In a quantitative finance context, we may wish to estimate the future price of an equity based on use of a GBM equation simulated M times.

For instance the process could be as follows for calculating a call option derivative based on estimate the price of a stock using Monte Carlo simulation:

1. Generate M different "trajectories" for the stock using a GBM simulation from time $t = 0$ to time $t = T$. This generates a set of N price estimates for M different simulations, with the notation:

$$\{S_i^j\} \qquad i = 0...N, \qquad j = 1...M \tag{4}$$

   This produces a vector of M values for S_T,

$$\{S_T^i\} \qquad i = 0...M \tag{5}$$

2. We next compute the pay off for each stock value. This is given by

$$H(S_T^i) \qquad i = 1...M \tag{6}$$

   Where

$$H(S_T) = max(S_T - K, 0) \tag{7}$$

   and K is the actual price of the equity or premium. The expected pay off can then be computed by the average of all pay offs:

$$E[H(S_T^i)] = \frac{1}{M}\Sigma_{i=1}^{M} H(S_T^i) \tag{8}$$

3. The item should then be discounted to present value, by either applying a discount factor DF_T or

$$\pi = e^{-rT} \times E[H(S_T)] \tag{9}$$

   Where π is the value of the derivative.

Binomial Trees [4]

[4] A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014

This approach builds a tree of possible prices. At each stage, the underlying can be assumed to go up or down by a given amount. The amount of change up (u) or down (d) is described by

$$u = e^{\sigma\sqrt{\Delta t}} \tag{10}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{11}$$

The probability of an equity going up p is

$$p = \frac{e^{r\Delta t} - d}{u - d} \tag{12}$$

The probability of the underlying going down is $1 - p$. The binomial tree is then built in the following phases:

1. Construct a tree where each level corresponds to a time step in the simulation period from $t = 0$ to $t = T$. For example in two simulation steps,

$$At = 0, S = S_0 \tag{13}$$

$$At = t_1, S = uS_0, \qquad dS_0 \tag{14}$$

$$At = t_2, S = u^2S_0, \qquad udS_0, \qquad udS_0, \qquad d^2S_0 \tag{15}$$

   Note that the central value at the last period is shared between adjacent nodes, hence there are only three distinct estimates after $N = 2$ steps.

   This produces a number of prices at each time step. The notation is based on the estimate number k at time T. There are N time steps

$$\{S_T^k\} \qquad k = 1...N + 1 \tag{16}$$

2. Once the tree has been built, the payoff $H(S_T^k)$ should be calculated for each S_T^k

3. Finally the tree is traversed back up towards the root node, calculating the discounted weighted probability of each node. If we define a given node (not a leaf, i.e. $t \neq T$) it has two children, one for the up price (denoted S_T^u) and one for the down price, S_T^d. The value for the parent node is given by

$$V_{T-1}^k = e^{-r\Delta t}[pH(S_T^u) + (1 - p)H(S_T^d)] \tag{17}$$

   in this case V_T^l is shorthand for $H(S_T^k)$

4. Once the tree has been traversed back to the top, the value of the derivative $\pi = V_1^1$

Finite Difference Method

The finite difference method is a method for discretising a differential equation.[5] In the quantitative finance approach, we want to discretise partial differential equations (PDEs). The finite differences method is based on the relationship:

[5] A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014

$$f(x) = \frac{df}{dx} \approx \frac{\Delta f}{\Delta x} = \frac{f_{i+1} - f_i}{\Delta x} \tag{18}$$

The most important PDE in finance is the *Black-Scholes PDE*, which is given by:

$$\frac{\delta V}{\delta t} + \frac{1}{2}\sigma^2 S^2 \frac{\delta^2 V}{\delta S^2} + rS\frac{\delta V}{\delta S} - rV = 0 \tag{19}$$

This is usually solved in the S and t axes, where $S \in [a, b]$ and $t \in [0, T]$. The domain of this equation is said to be

$$\Omega = \{(S, t)\forall S \in [a, b] \times t \in [0, T]\} \tag{20}$$

In other words, as the finite difference method is solving some partial differential equation in S and t, the solution space is the rectangular domain defined by the ranges of S and t. For a European call,

$$V(S, T) = max(S - K, 0) \tag{21}$$

The boundary conditions are $V(a, t) = 0$ and $V(b, t) = S$. This equation can be transformed with some variable substitution so that

$$\frac{\delta u}{\delta \tau} = \frac{\delta^2 u}{\delta x} \qquad -\infty < x < \infty, \tau > 0 \tag{22}$$

This is a dimensionless PDE with a new solution domain $\Omega = \{(x, \tau)\}$. The payoff relationship therefore becomes [6]

[6] Where $k = \frac{r}{0.5 \times \sigma^2}$

$$u(x, 0) = max(e^{\frac{1}{2}(k+1)x} - e^{\frac{1}{2}(k-1)x}, 0) \tag{23}$$

Using finite differences, the return can be described as[7]

[7] Where $\alpha = \frac{\Delta\tau}{(\Delta x)^2}$

$$u_{i,j+1} = \alpha u_{i+1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i-1,j} \tag{24}$$

This relationship can be solved iteratively, using the following steps:

1. Discretise the domain into N space divisions of dS and M time divisions of dT. Use these to determine the steps $\Delta\tau, \Delta x$.
2. Use finite differences to approximate the derivatives
3. Calculate the results of the equation iteratively for each time step

For a worked example, see [8], end of chapter 3.

[8] A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014

Bibliography

A. Pena. *Advanced Quantative Finance With C++*. Packt Publishing, 2014.

Nicholas Nasim Taleb. *Fooled By Randomness*. Penguin, 2005.

Index