Monte Carlo Simulation

Goals

- Apply the Monte Carlo method to option pricing
- Developing algorithms in C++ and library integration
- Advantages and disadvantages of Monte Carlo
- Pointers to more advanced applications

History of Monte Carlo

- Invented during WWII (John von Neumann)
- First applied to option pricing by Phelim Boyle (1976)
- It is popular because it always produces some kind of answer
- Applicable to n-factor problems

Basic Idea of Monte Carlo

- Simulate the underlying's SDE for t = 0 (now) to t = T (expiry) NSIM times
- We normally have to simulate the SDE using the finite difference method (FDM)
- Compute the payoff at t = T for each simulation; average all the payoffs over NSIM
- Apply discounting from t = T to t = 0 to the average
- (Clewlow/Strickland 1998: "Implementing Derivatives Models", Wiley)

‘Building Block’ Classes

- Classes, structs or namespaces for SDE, FDM, RNG
- Choose between home-grown RNG and Boost RNG
- The algorithmic code that ‘ties in’ the building blocks
- Configuring the application, initialising the data etc.

5

What Kinds of Solutions?

- Determined by accuracy, efficiency and functionality requirements
- ‘get it working’ (approach by Clewlow/Strickland)
- ‘get it right’ (adaptable, Kienitz/Duffy 2010)
- ‘get it optimised’ (Monte Carlo engines and production software)

6

Random Number Generators

- We need to generate Gaussian (normal) random variables
- Usually we get them from uniform random numbers
- Methods: Box-Muller, Polar Marsaglia, Mersenne Twister, lagged Fibonacci
- Lots of code floating on internet

SDES (1/2)

- We concentrate one-factor linear and nonlinear Geometric Brownian Motion (GBM)

$$dS_t = (r - D)S_t dt + \sigma S_t dW_t \; (S_t \equiv S(t))$$

r =(constant) interest rate.

D =constant dividend.

σ =constant volatility.

dW_t =increments of the Wiener process.

- Methods can be applied to mean-reverting SDE

SDES (2/2)

$$dr = \kappa(\theta - r)\,dt + \sigma r^{\beta} dw$$

$r = r(t)$ = level of short rate at time t.
dW = increment of a Wiener process.
θ = long-term level of r.
κ = speed of mean reversion.
σ = volatility of the short rate.
$\beta = 0$ for Vasicek model, $\frac{1}{2}$ for CIR model.

9

Finite Difference Approximations

- Many choices
- We need accurate and stable schemes (easier said than done)
- (Explicit and Implicit) Euler, (Ito) Milstein, Runge-Kutta, Heun, semi-implicit
- Not well-developed; lots of experimentation needed

10

Prototype SDE

- Nonlinear, autonomous SDE

$$dX(t) = \mu(X(t))dt + \sigma(X(t))dW(t) \quad 0 < t \leq T$$
$$X(0) = A$$

- Special case is linear GBM SDE
- Many other kinds of SDE

FDM Schemes

- Explicit Euler

$$X_{n+1} = X_n + \mu_n \Delta t + \sigma_n \Delta W_n$$

- Milstein

$$X_{n+1} = X_n + \mu_n \Delta t + \sigma_n \Delta W_n + \frac{1}{2}[\sigma' \sigma]_n((\Delta W_n)^2 - \Delta t)$$

- Semi-implicit Euler

$$X_{n+1} = X_n + [\alpha\mu_{n+1} + (1-\alpha)\mu_n]\Delta t + \sigma_n \Delta W_n$$

$$\alpha = \tfrac{1}{2} \text{ (Trapezoidal)}, \alpha = 1 \text{ (Backward Euler)}$$

$$(\sigma^1 \equiv \tfrac{d\sigma}{dx})$$

The Algorithm

For each $j = 1, .., M$ calculate$(M == \text{NSIM})$

$$C_{T,j} = \max(0, S_{T,j} - K)$$

and

$$\hat{C} = \exp(-rT)\frac{1}{M}\sum_{j=1}^{M} \max(0, S_{T,j} - K)$$

Then $\hat{C}$ is the desired call price.

13

Advantages of MC

- Applicable to a wide range of problems
- Easy to understand and to program
- Well-established in the marketplace
- Can be used as a ‘second opinion’ for other methods (for example, FDM)

14

Disadvantages of MC

- Applicability breaks down at some stage (option sensitivities, early exercise feature)
- Lack of predictable accuracy
- Slow (can be speedup by a combination of hardware and software)
- Takes some effort to make MC code easy to analyse (reporting algorithm progress, reporting)

N-Factor Problems

- MC is applicable to these problems
- We need to generate correlated random numbers
- FD schemes can be extended to N-factor problems
- Could be a project for later