UPPSALA
UNIVERSITET

ASSIGNMENT 4 - EXOTIC OPTIONS

MONTE CARLO METHODS TO PRICE CHOOSER OPTIONS

INTRODUCTION

Our task is to calculate the price of a chooser option using Quasi-Monte Carlo method. In this presentation we'd like to discuss the following points:

- what a chooser option is and how its price can be expressed using the Black-Scholes approach
- how its price can be calculated using Monte Carlo simulations
- comparing the Δ of call, put and chooser options of European style
- what quasi-random sequences and quasi-Monte Carlo methods are
- how ordinary and quasi-Monte Carlo methods compare when calculating the price of a chooser option

CHOOSER OPTION

A chooser option

- gives holder a right to decide if it's a call or put at a specific time T_c, $0 < T_c < T$
- European style mostly
- is more expensive than a comparable vanilla option due to its greater flexibility

Chooser option: Black-Scholes approach

Note that at T_c, the arbitrage-free price of the chooser option will be:

$$V(S_{Tc}, T_c) = \max(C(S_{T_c}; T - T_c; K), P(S_{T_c}; T - T_c; K)) \tag{1}$$

where

- K - strike price,
- S_{T_c} stock price at T_c (unknown/random)
- $C(s; t; K)$, $P(s; t; K)$ - European call/put price with time to maturity T, current stock price s, strike price K.

By Black-Scholes, the price at $t = 0$ is

$$e^{-rT_c}\mathrm{E}[V(S_{T_c}, T_c)].$$

Can be calculated explicitly (Rubinstein, 1991) or using simulations.

UPPSALA
UNIVERSITET

Chooser option: Monte Carlo approach

Let us calculate

$$V_{cho}(s) = e^{-rT_c}\mathrm{E}[V(S_{T_c}, T_c)],$$

numerically, using Monte-Carlo simulations.
In our assignment, S_t is driven by

$$dS_t = rS_t dt + \sigma S_t dW_t,\, S_0 = s.$$

S_t is Geometric Brownian Motion, and can be explicitly expressed as

$$S_t = S_0 \cdot \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right),$$

Simulating $W(t)$ is easy:

$$W(t) \sim \sqrt{t} \cdot \mathcal{N}(0,1).$$

IMPLEMENTATION OF REGULAR CHOOSER MONTE-CARLO METHOD

The implementation is generally unchanged from the vanilla call/put option one, thus making the it relatively simple

1. Implementation based on MC MATLAB-function for vanilla options
2. At chosen time t, the price S_t is stored in a variable
3. Numerical solution according to equation 1
4. Analytical solution obtained from blsprice (Matlab Financial Toolbox)
5. (Error calculation if wanted)

COMPUTATIONS OF $\Delta = \frac{\partial V}{\partial S}$

Numerical calculations of Δ requires approximating ∂V_{cho} and ∂S. Using central differance

$$\partial V = V_{cho}(S_0 + dS) - V_{cho}(S_0 - dS) \qquad \partial S = 2dS$$

and calculating $\frac{\partial V}{\partial S}$ results in the formula for approximating Δ for a chooser using CD

$$\Delta = \frac{V_{cho}(S_0 + dS) - V_{cho}(S_0 - dS)}{2dS} \tag{2}$$

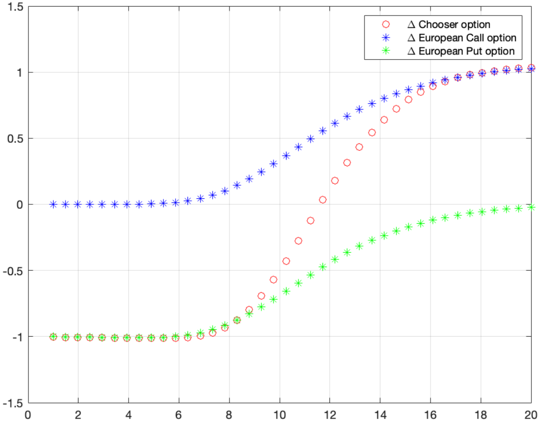
Using equation 2, Δ can be calculated for varying initial stock prices S_0 using the following pseudo code

UPPSALA UNIVERSITET

COMPUTATIONS OF $\Delta \frac{\partial V}{\partial S}$

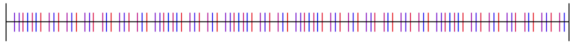
$S_0 = \text{array}(S_{0_{min}} : n : S_{0_{max}})$
$delta = array(length(S_0))$
set dS

for i in S_0 **do**
$V_1 = F_{cho}(S_0(i) + dS)$
$V_2 = F_{cho}(S_0(i) - dS)$
$delta(i) = (V_1 - V_2)/2dS$
end for

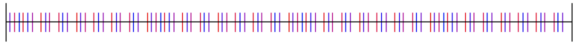


Quasi vs Pseudo Monte-Carlo method

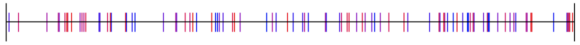
Sobol Sequence



Halton Sequence



Pseudo random numbers



DISCREPANCY

- Discrepancy is a measurement of the highest and lowest density of points in a sequence.
- High discrepancy means that there is either a large area that has a large area of empty spaces, or that there is an area that has a high density of points.
- Low discrepancy means that there are neither, and that your points are more or less pretty evenly distributed.

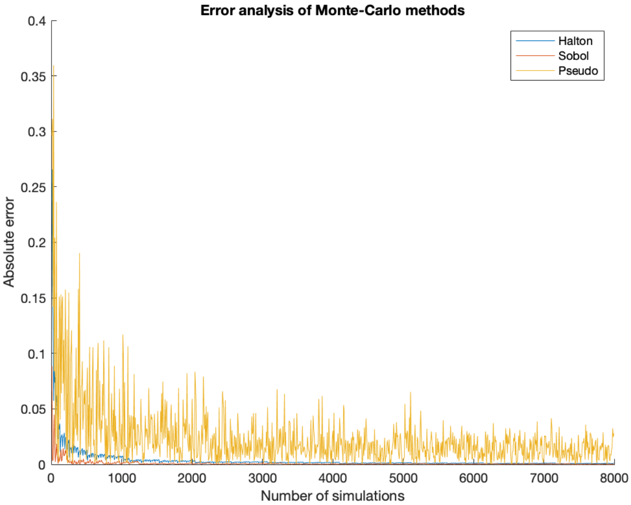
Quasi vs Pseudo Monte-Carlo method - Advantages

1. Quasi sequences yields faster convergence rate compared to pseudo random numbers
2. A use of low discrepancy sequences gives rise to deterministic error bounds instead of probabilistic as in ordinary Monte Carlo
3. Lower discrepancy leads to lower variance.

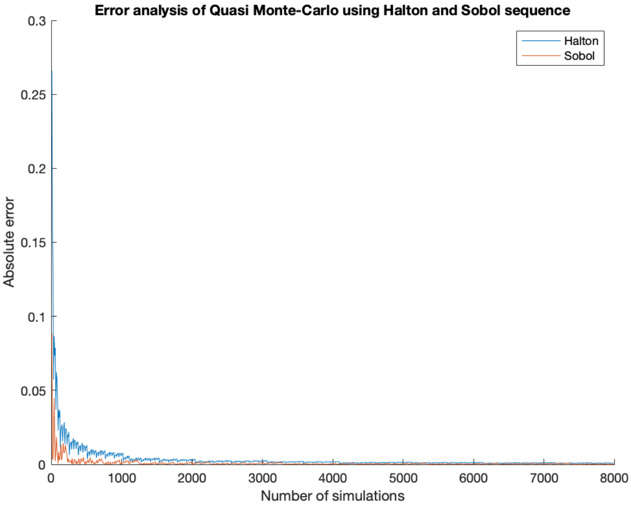
Quasi vs Pseudo Monte-Carlo method - Implementation

1. Create a stream reference object of Halton or Sobol sequence (*MATLAB: qrandstream*).
    1.1 Skip and leap to decrease discrepancy.
2. Generate a one dimensional array containing quasi-random points from the stream (*MATLAB: qrand*).
3. Inverse the matrix to obtain the inverse cumulative distribution of the normal distribution (*MATLAB: norminv*).

Quasi vs Pseudo Monte-Carlo method



Quasi vs Pseudo Monte-Carlo method - Sobol vs Halton



UPPSALA UNIVERSITET

Quasi vs Pseudo Monte-Carlo method - Results

Pseudo	Sobol	Halton
0.022879	0.00092249	0.0037476

Table: Mean error for the Monte-Carlo methods

Pseudo	Sobol	Halton
0.48302	0.71288	0.88208

Table: Convergence study of the Monte-Carlo methods

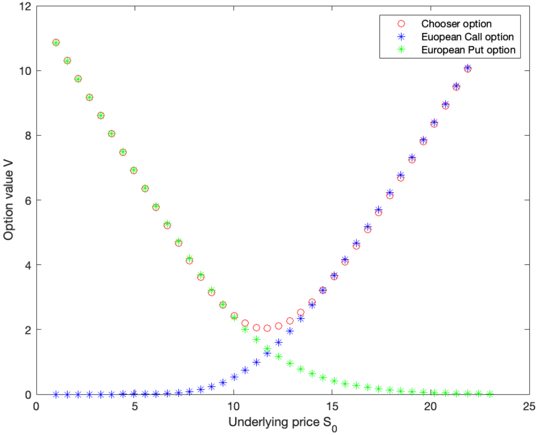
UPPSALA
UNIVERSITET

Quasi vs Pseudo Monte-Carlo method - Conclusion

1. Overall Monte-Carlo is a useful method to price path-dependent options
2. More simulations yield lower variance and higher pricing approximation accuracy
3. Implementing Quasi sequences into Monte-Carlo successfully increases the convergence rate
4. Quasi Monte-Carlo shows less variance compared to pseudo Monte-Carlo

NUMERICAL RESULT

Option price V for varying underlying prices S_0 using K = 12



UPPSALA UNIVERSITET

CONCLUSION

1. We've seen that quasi-Monte Carlo methods allow to significantly decrease computational time compare to usual Monte Carlo simulations
2. Quasi-Monte Carlo method allows to achieve better convergence rate
3. For large N the accuracy of the quasi-Monte Carlo method increases faster than that of the Monte Carlo method
4. A use of low-discrepancy sequences gives rise to deterministic error bounds instead of probabilistic as in ordinary Monte Carlo

Further improvement can be achieved by using so-called randomized quasi-Monte Carlo method which combines ordinary and quasi-Monte Carlo methods.

Thank you for your attention!