UPPSALA
UNIVERSITET

Institutionen för informationsteknologi

Besöksadress:
hus 10, Lägerhyddsvägen 1

Postadress:
Box 337
SE-751 05 UPPSALA

Telefon:
018–471 0000 (växel)

Telefax:
018-471 20 00

Hemsida:
http:/www.it.uu.se/

Department of Information Technology

Visiting address:
hus 10, Lägerhyddsvägen 1

Postal address:
Box 337
SE-751 05 Uppsala
SWEDEN

Telephone:
+46 18–471 0000 (switch)

Telefax:
+46 18-471 20 00

Web page:
http:/www.it.uu.se/

Assignment - Exotic options

Your task is to implement a solver to price an exotic option. The code shall compute the value of the option and also the value of the Greek Δ. Each group (1-3 students) is assigned a specific type of option and numerical method. The options considered are of European type and the underlying asset follows a geometric Brownian motion and pays no dividend. The parameters r and σ are constant in both "space" and time.

The result shall be presented as a short lecture to the rest of the class on October 5. You are encouraged to use a pdf- or ppt-presentation but you may have other interesting suggestions on how to present your results. Before you present the lecture to the class you will send your presentation to Elisabeth Larsson for feedback. This shall be done no later than October 2, such that you have time for potential corrections. The lecture shall at least include:

- The type of option you have studied.
- The numerical method you have used and how the solver for the vanilla (standard) European option had to be changed, and what new difficulties you encountered.
- The computation of Δ - how was this accomplished?
- Your numerical results.
- Discussion/Conclusions

You have the opportunity to have 10 min tutoring/group. Reserve your time slot for this here https://doodle.com/meeting/participate/id/e7LJ7l1a (10 min tutoring/group).

Finally, you have to attend the other presentations and answer a questionnaire. The questionnaire is open in the student portal October 5–October 7.

1 Asian options

The Asian option is a path-dependent option where the value depends on the average price of the underlying. In case of the continuous arithmetic average Asian option, the average is given by

$$A_t = \frac{1}{t}\int_0^t S_\tau d\tau. \tag{1}$$

The payoff is either

$$V_T = (A_T - K)^+, \quad \text{(fixed strike), or} \tag{2}$$

$$V_T = (S_T - A_T)^+, \quad \text{(floating strike).} \tag{3}$$

1.1 Assignment 4 a- Finite differences to price Asian options

Implement the floating strike Asian option using the reduction to 1D outlined in Seydel. Use a second order upwind stencil for the first derivative and BDF-2 (Backward Differentiation Formula, 2nd order) for time-stepping.

Reference:
Seydel R., *Tools for Computational Finance*, 4.ed., Springer, Berlin (2009), pp.240–244.

1.2 Assignment 4 b - Monte-Carlo methods to price Asian options

Implement both the floating strike and the fixed strike Asian options using the standard Monte Carlo method. To reduce the variance for the fixed strike Asian option, use the method of control variates with the geometric average fixed strike option as the control.

Reference:
Kemna and Vorst, *A Pricing Method for Options Based On Average Asset Values*, Journal of Banking and Finance 14 (1990), pp. 113–129.

2 Barrier options

For a barrier option, the contract either becomes void (out option) or comes into existence (in option) as the price of the underlying reaches a barrier. In the case of an up-and-out call option, the payoff is given by

$$V_T = \begin{cases} (S-K)^+, & S(t) < B \text{ for all } t \in [0,T] \\ 0, & S(t) \geq B \text{ for some } t \in [0,T], \end{cases} \tag{4}$$

where B is the barrier.

Reference:
Wilmott P., *Paul Wilmott introduces quantitative finance*, 2. ed., John Wiley & Sons Ltd (2007), p.308.

2.1 Assignment 4 c - Finite differences to price barrier options

Modify your finite difference implementation of the Black–Scholes equation with appropriate boundary conditions for the up-and-out call option. Use BDF-2 (Backward Differentiation Formula, 2nd order) for time-stepping. Study the effect of varying the volatility.

2.2 Assignment 4 d - Monte-Carlo methods to price barrier options

Modify your Monte Carlo code to include the up-and-out barrier. Study the convergence with respect to the time step. Implement the conditional exit probability method for reducing the bias. Study the effect of varying the volatility for S_0 close to the barrier, and far away from it.

Reference:
Moon K.-S., *Efficient Monte Carlo Algorithm for Pricing Barrier Options*, Commun. Korean Math. Soc. 23 (2008), No. 2, pp. 285-294.

3 Binary options

Two examples of binary options are the cash-or-nothing option, where the payoff is

$$V_T = \begin{cases} C, & S_T \geq K, \\ 0, & S_T < K, \end{cases} \tag{5}$$

and the asset-or-nothing option, where the payoff is given by

$$V_T = \begin{cases} S_T, & S_T \geq K, \\ 0, & S_T < K. \end{cases} \tag{6}$$

Reference:

Hull J., *Options, futures, and other derivatives*, 7. ed. Upper Saddle River, N.J., Pearson Prentice Hall (2009).

3.1 Assignment 4 e - Finite differences to price binary options

Derive the boundary conditions for the two binary options and modify your finite difference code to compute the value of these options. Use BDF-2 (Backward Differentiation Formula, 2nd order) for time-stepping. Compute the convergence order in space and time. Is it second order? Why/why not?

3.2 Assignment 4 f - Monte-Carlo methods to price binary options

Modify your Monte Carlo code to compute the value of these two binary options. When using exact time integration, a higher convergence rate can be achieved by using a quasi-Monte Carlo method, where low discrepancy numbers replace the random ones. Implement a quasi-Monte Carlo method using either Halton or Sobol points. For generating these points, see the help for 'qrandstream' and 'norminv' in MATLAB.

4 Compound options

A compound option is an option on an option. With other words, you buy an option with strike price K_1 and time to maturity t_1, which gives you the right to buy (in the case of a call) or sell (in the case of a put) another option (call or put) with strike price K_2 which expires at time $t_2 > t_1$.

There are four basic cases, Call on Call (CoC), Call on Put (Caput), Put on Call (PoC) and Put on Put (PoP).

Reference:

Robert Geske, *The valuation of compound options*, Journal of Financial Economics 7 (1979) 63-68.

4.1 Assignment 4 g - Finite Differences to price compound options

Derive the boundary conditions and final values and modify your finite difference code to compute the value of compound options. Use BDF-2 (Backward Differentiation Formula, 2nd order) for time-stepping.

4.2 Assignment 4 h - Monte-Carlo methods to price compound options

Modify your Monte-Carlo code to price a compound option using Monte Carlo. Note that the SDE governing the price of the stock can be integrated exactly.

When using exact time integration, a higher convergence rate can be achieved by using a quasi-Monte Carlo method, where low discrepancy numbers replace the random ones. Implement a quasi-Monte Carlo method using either Halton or Sobol points. For generating these points, see the help for 'qrandstream' and 'norminv' in MATLAB.

5 Chooser options

A simple chooser option gives the purchaser the right to decide at a later time t whether the option should be a European call or put with strike price K and expiration time $T > t$. A generalization would be to let the call and put have different strike prices and expiration times, in which case we call it a complex chooser option, but for this assignment we only consider simple chooser options.

Assuming the stock price follows a geometric brownian motion, the payoff V is governed by the standard Black-Scholes equation. An analytic formula for the price of a chooser option can be derived, and is found in for instance "The complete guide to option pricing formulas" by Espen Gaarder Haug (available through Google Books).

5.1 Assignment 4 i - Finite differences to price chooser options

Derive the boundary conditions and final value and modify your finite difference code to compute the value of chooser options. Use BDF-2 (Backward Differentiation Formula, 2nd order) for time-stepping.

5.2 Assignment 4 j - Monte-Carlo methods to price chooser options

Modify your Monte-Carlo code to price a chooser option using Monte Carlo. Note that the SDE governing the price of the stock can be integrated exactly.

When using exact time integration, a higher convergence rate can be achieved by using a quasi-Monte Carlo method, where low discrepancy numbers replace the random ones. Implement a quasi-Monte Carlo method using either Halton or Sobol points. For generating these points, see the help for ‘qrandstream’ and ‘norminv’ in MATLAB.