MATHEMATISCHES INSTITUT
DER UNIVERSITÄT ZU KÖLN
Prof. Dr. Rüdiger Seydel
Dipl.-Math. Christian Jonen

CCFinance
Cologne Computational Finance Laboratory

Winter 2009/10
October 19

Computational Finance 2 - 2nd Assignment

Deadline: October 26

Exercise 2 (Cubic B-Spline) (4+1+1 Points)

Suppose an equidistant partition of an interval be given with mesh-size $h = x_{k+1} - x_k$. Cubic B-Splines have a support of four subintervals. In each subinterval the spline is a piece of polynomial of degree three. Apart from special boundary splines, the cubic B-splines φ_i are determined by the requirements

$$\begin{aligned} \varphi_i(x_i) &= 1 \\ \varphi_i(x) &\equiv 0 \quad \text{for } x < x_{i-2} \\ \varphi_i(x) &\equiv 0 \quad \text{for } x > x_{i+2} \\ \varphi &\in \mathcal{C}^2(-\infty, \infty). \end{aligned}$$

To construct the φ_i proceed as follows:

a) Construct a spline $S(x)$ that satisfies the above requirements for the special nodes

$$\widetilde{x}_k := -2 + k \quad \text{for } k = 0, ..., 4.$$

b) Find a transformation $T_i(x)$, such that $\varphi_i = S(T_i(x))$ satisfies the requirements for the original nodes.

c) For which i, j does $\varphi_i \varphi_j = 0$ hold?

Exercise 3 (Finite-Element Matrices) (9 Points)

For the *hat functions* φ introduced in the lecture calculate for arbitrary subinterval $\mathcal{D}_k$ all nonzero intergrals of the form

$$\int \varphi_i \varphi_j dx, \quad \int \varphi_i' \varphi_j dx, \quad \int \varphi_i' \varphi_j' dx$$

and represent them as local 2×2 matrices.

Exercise 4 (Three-Dimensional Binomial Tree) (3+14+3 Points)

In order to price options on two assets, the classical binomial tree can be extended to a three-dimensional binomial tree, see figure 1. The idea is to allow for movements of share 1 and share 2. So, the asset price vector $S_t = (S_t^1, S_t^2)$ at time t can move to four different states $S_{t+1}^{(1)}$, $S_{t+1}^{(2)}$, $S_{t+1}^{(3)}$ and $S_{t+1}^{(4)}$ at time $t + 1$.

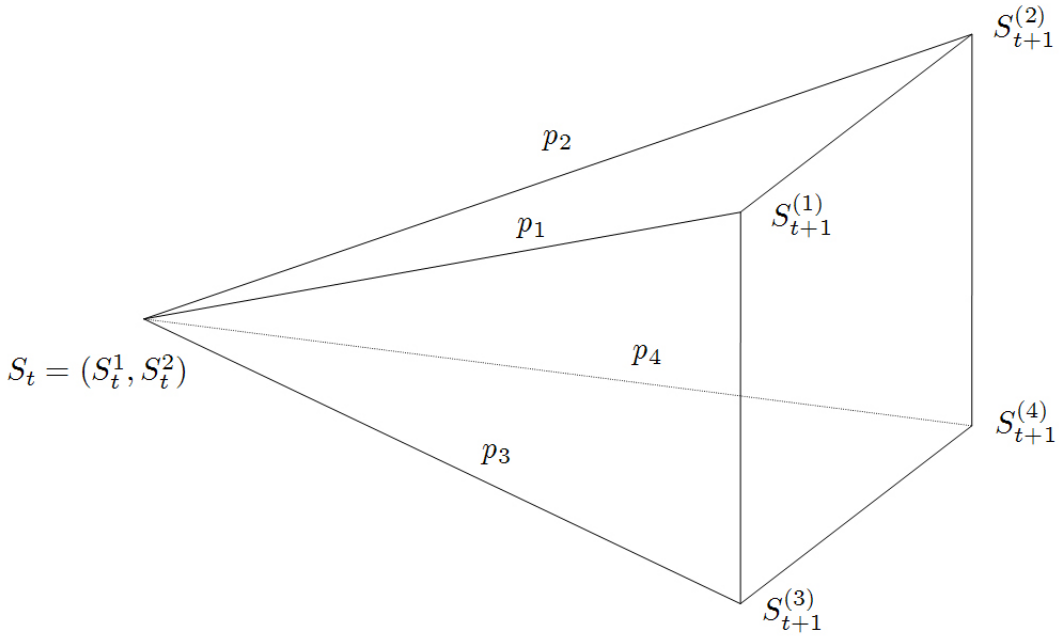

Figure 1: Structure of a three-dimensional binomial tree.

Table 1: Several payoffs of rainbow call and put options on two shares. The notation $(x)^+$ means $\max(0, x)$.

name	call	put
maximum (max)	$(\max(S^1, S^2) - K)^+$	$(K\text{-}\max(S^1, S^2))^+$
minimum (min)	$(\min(S^1, S^2) - K)^+$	$(K\text{-}\min(S^1, S^2))^+$
geometric average	$((\prod_{d=1}^{2} S^d)^{1/2} - K)^+$	$(K - (\prod_{d=1}^{2} S^d)^{1/2})^+$
arithmetic average	$(\frac{1}{2}\sum_{d=1}^{2} S^d - K)^+$	$(K - \frac{1}{2}\sum_{d=1}^{2} S^d)^+$

a) Several payoffs of rainbow options are listed in table 1, and figure 2 shows the payoff of a maximum call option on two shares. Plot the payoff of a geometric average put option with strike $K = 40$.

b) There are several approaches for realizing a three-dimensional binomial tree. By constructing such a tree, the correlation between share 1 and share 2 has to take into account. In 1994 Rubinstein proposed the approach illustrated in figure 3 with the following settings:

$$
\begin{aligned}
u &= \exp\left((r - \delta_1 - \sigma_1^2/2)\Delta t + \sigma_1\sqrt{dt}\right) \\
d &= \exp\left((r - \delta_1 - \sigma_1^2/2)\Delta t - \sigma_1\sqrt{dt}\right) \\
A &= \exp\left((r - \delta_2 - \sigma_2^2/2)\Delta t + \sigma_2\sqrt{\Delta t}(\rho + \sqrt{1-\rho^2})\right) \\
B &= \exp\left((r - \delta_2 - \sigma_2^2/2)\Delta t + \sigma_2\sqrt{\Delta t}(\rho - \sqrt{1-\rho^2})\right) \\
C &= \exp\left((r - \delta_2 - \sigma_2^2/2)\Delta t - \sigma_2\sqrt{\Delta t}(\rho - \sqrt{1-\rho^2})\right) \\
D &= \exp\left((r - \delta_2 - \sigma_2^2/2)\Delta t - \sigma_2\sqrt{\Delta t}(\rho + \sqrt{1-\rho^2})\right),
\end{aligned}
$$

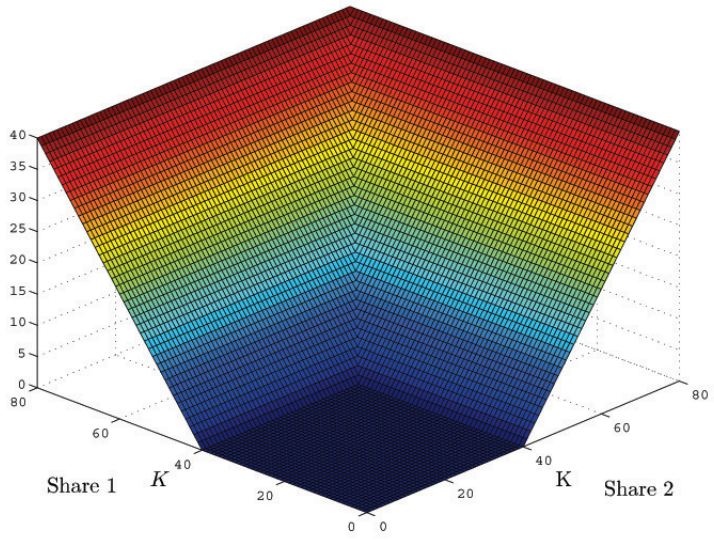

Figure 2: Payoff of a maximum call option on two shares with strike K=40.

see [Rub94]. Then, the probability to reach one of four states in the next time step is $p \equiv 0.25$.

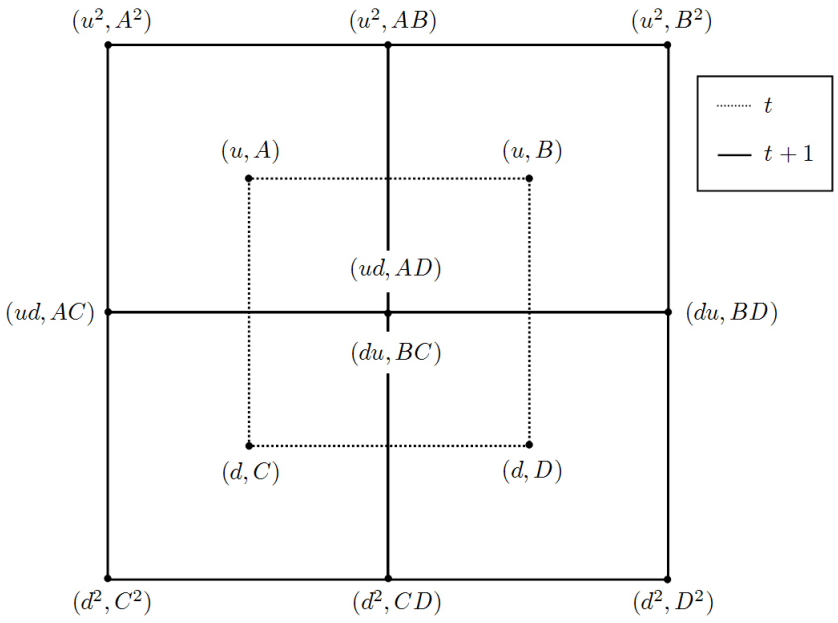

Figure 3: Slice plane of the Rubinstein binomial tree for pricing options on two shares. Two time steps are illustrated.

Implement the Rubinstein approach in C, C++, Java or Fortran. To this end, write a subroutine which calculates the fair price of an option on two shares. The subroutine should have the following input arguments:

r (interest rate), S_0^1 (initial value of share 1), S_0^2 (initial value of share 2), δ_1 (dividend yield of share 1), δ_2 (dividend yield of share 2), σ_1 (vola of share 1), σ_2 (vola of share 2), T (maturity), K (strike), $type1$ (European/American option), $type2$ (put/call), $type3$ (type of rainbow option), N (number of time steps).

Proceed as follows:

- **Initializing at maturity T:** Calculate the share prices at maturity T and initialize a matrix to save the option values for each state. For this purpose, implement the following loops:

  for $i = 0, ..., N$
  $S_1 = S_0^1 u^i d^{N-i}$
  for $j = 0, ..., N$
  $S_2 = S_0^2 \exp\left((r - \delta_2 - \sigma_2^2/2)Ndt + \sigma_2(\rho(2i - N)\sqrt{dt} + \sqrt{1-\rho^2}(2j - N)\sqrt{dt})\right)$
  V_{ij} =payoff$(S_1, S_2, type1, type2, type3, K)$

- **Time loop:** Implement an outer loop, namely the time loop, combined with the following inner loops:

  for $i = 0, ..., t$ // t (integer value) denotes the current time step
  $S_1 = S_0^1 u^i d^{t-i}$
  for $j = 0, ..., t$
  $S_2 = S_0^2 \exp\left((r - \delta_2 - \sigma_2^2/2)tdt + \sigma_2(\rho(2i - t)\sqrt{dt} + \sqrt{1-\rho^2}(2j - t)\sqrt{dt})\right)$
  $V_{ij} = \exp(-rdt)\ p(V_{ij} + V_{i+1,j} + V_{i,j+1} + V_{i+1,j+1})$

  By doing so, your code should be able to price American options as well.

- **Option Price at** $t_0(= 0)$**:** $V(0, 0)$

c) In order to test the correctness of your code, price an European max call option with the following settings:

$$r = 0.04879, S_0^1 = 40, S_0^2 = 40, \delta_1 = \delta_2 = 0.0, \sigma_1 = 0.2, \sigma_2 = 0.3, \rho = 0.5, T = 7/12. \tag{1}$$

Table 2: Prices of an European max call option on 2 shares with the settings (1) for three different strike prices.

	K=35	K=40	K=45
price	9.420	5.488	2.795

Assuming that table 2 shows the correct values of this option for three different strike prices, calculate the absolute errors for $N = 10$, 50, 100, 800 time steps. By the way, by using $N = 800$ time steps and the settings (1), the price of an American max put option is 1.46776 for K=40.

Information:

Deadline for Programming Assignment (Exercise 4): November, 2

References:

[Rub94] M. Rubinstein, *Return to Oz*, Risk 11 (1994), pp. 67-71.