Computational Methods in Finance
Computational project 1

Matej Rojec, Patrycja Ozgowicz, Erik Hamberg

April 16, 2023

Contents

1 Task 1 **3**

List of Figures

1 Plot of $V(S, t)$ in the case of a European call option, obtained by explicit method. We took $N_S = 50$ and $N_t = 1000$. 5
2 Plot of $V(S, t)$ in the case of a European call option, obtained by Crank-Nicolson method. We took $N_S = 50$ and $N_t = 1000$. . . . 6
3 Plot of $V(S, t)$ in the case of a European call option, obtained from the Black-Scholes formula. 7
4 Errors for the two method. 8

List of Tables

1 Errors calculated for the explicit method at time $t = 1$. 9
2 Errors calculated for the Crank-Nicolson method at time $t = 1$. . 9
3 Errors calculated for the explicit method at time $t = 2$. 10
4 Errors calculated for the Crank-Nicolson method at time $t = 2$. . 10
5 Errors calculated for the explicit method at time $t = 3$. 10
6 Errors calculated for the Crank-Nicolson method at time $t = 3$. . 10

1 Task 1

We consider the region

$$\mathcal{R}_V^T = (S,t), 0 < S < S^*, 0 \leq t \leq T, \tag{1}$$

for a sufficiently large S^* and the terminal/boundary value problem for the Black-Scholes equation defining $V(S,t)$ to be the value of an option at the point (S,t). In order to replace the terminal value problem associated with the Black-Scholes equation with an initial value problem, we perform a change of variables $U(S,t) := V(S,T-t)$ and consider the problem

$$\begin{cases} \frac{\partial U}{\partial t} = \frac{\sigma^2}{2}S^2\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU & \text{in } \mathcal{R}_V^T \\ U(S,0) = u_0(S) & S \in [0,S^*] \\ U(0,t) = u_a(t) & t \in [0,T] \\ U(S^*,t) = u_b(t) & t \in [0,T] \end{cases} \tag{2}$$

for some functions u_0, u_a, and u_b that depend on the type of option and are assumed to be known.

a) Write Matlab routines for solving problem (2) using the explicit method and also using the Crank-Nicolson (CN) method.

The code can be found in the zip file attached in the email. This is done in the Matlab files `BS-explicit.m` and `BS-CN.m` files. We will initialize both solutions the same way. That is we instantiate variables for the risk-free interest rate, r, the volatility, σ, the strike price, K, the lower spacial bound, a, the upper price limit, S^*, the last time step, T, the number of spatial and time steps, N_s and N_t, and the step sizes, h_s and h_t.

```
r = 0.06;
sigma = 0.3;
K = 10;
a = 0;
S_star = 15;
T = 1;
NS = 100;
Nt = 1000;
hS = (S_star-a)/NS;
ht = T/Nt;
```

Then the solution space is defined as a grid. The boundaries are defined as well.

```
lS = linspace(a,S_star,NS);
lt = linspace(0,T,Nt);
grid = meshgrid(lS,lt);
```

```
u0 = @(S)(max(S-K,0));
ua = 0;
ub = @(t)(S_star - K*exp(-r*t));

grid(1,:) = u0(lS);
grid(:,1) = ua;
grid(:,end) = ub(lt);
```

After this we can calculate the solutions. To find the solution with the explicit method the following loop is run over the grid.

```
i = 2:NS-1;
for j=1:Nt-1
        grid(j+1, i) = ht/2 * ((sigma^2) * i.^2 + r*i) .* grid(j,i+1) +
        (1-sigma^2 * i.^2 * ht - r*ht) .*
        grid(j,i) + ht/2 * (sigma^2 * i.^2 - r*i) .* grid(j,i-1);
end
```

and to solve it with the Crank-Nicolson method we run the following code segment.

```
i = 1:NS-2;

a = -(sigma^2)/4 * ht * i.^2 + r*ht/4*i;
b = 1 +(sigma^2)/2 * ht * i.^2 + r*ht/2;
c = -(sigma^2)/4 * ht * i.^2 - r*ht/4*i;
d = 1 -(sigma^2)/2 * ht * i.^2 - r*ht/2;


A = zeros(NS-2);
A(1:NS-1:end) = b;
A(2:NS-1:end-1) = a(2:end);
A(NS-1:NS-1:end) = c(1:end-1);



B = zeros(NS-2);
B(1:NS-1:end) = d;
B(2:NS-1:end-1) = -a(2:end);
B(NS-1:NS-1:end) = -c(1:end-1);
first_col = zeros(1,NS-2);
last_col = zeros(1,NS-2);
first_col(1) = -a(1);
last_col(end) = -c(end);

B = [first_col' B last_col'];

for j=1:Nt-1
```

```
    vector = grid(j, :)';
    add = zeros(1,NS-2)';
    add(1) = -a(1)* grid(j+1,1);
    add(end) = -c(end)*grid(j+1, end);
    u_solution = A \(B * vector + add);
    grid(j+1, 2:end-1) = u_solution;
end
```

b) Calculate and plot the solutions of (2) obtained by both numerical methods in the case of a European call option, for the parameters $r = 0.06$, $\sigma = 0.3$, $T = 1$ and $K = 10$ and taking $S^* = 15$.

In this point, we consider the problem

$$\begin{cases} \frac{\partial U}{\partial t} = 0.045S^2\frac{\partial^2 U}{\partial S^2} + 0.06S\frac{\partial U}{\partial S} - 0.06U & \text{in } \mathcal{R}_V^T \\ U(S,0) = \max\{S_1 - 10, 0\} & S \in [0,15] \\ U(0,t) = 0 & t \in [0,1] \\ U(15,t) = 15 - 10e^{-0.06t} & t \in [0,1] \end{cases} \tag{3}$$

and $U(S,t) := V(S, T-t)$.

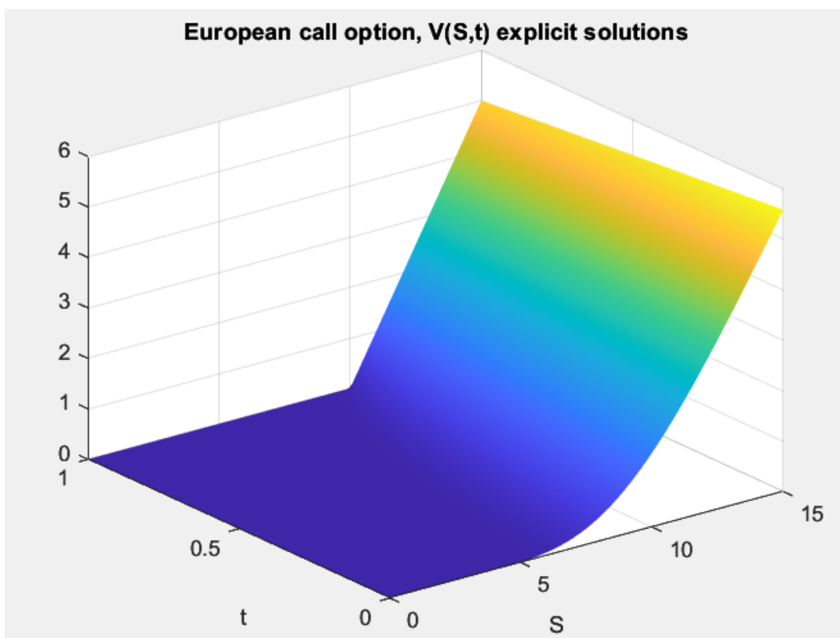


Figure 1: Plot of $V(S,t)$ in the case of a European call option, obtained by explicit method. We took $N_S = 50$ and $N_t = 1000$.

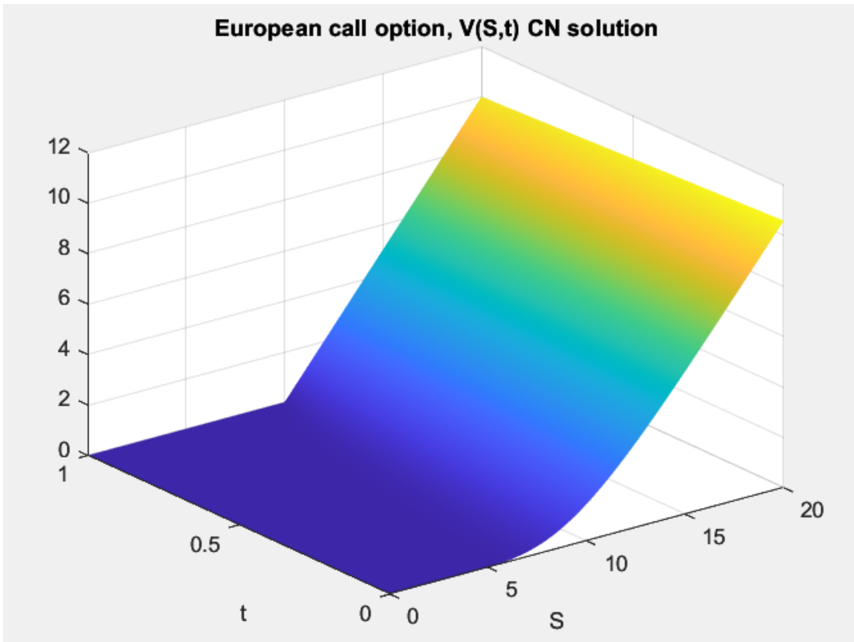


Figure 2: Plot of $V(S,t)$ in the case of a European call option, obtained by Crank-Nicolson method. We took $N_S = 50$ and $N_t = 1000$.

To generate the plots we use the following code.

```
figure(1)
surf(lS,lT-lt,grid, 'LineStyle','none', 'FaceColor','flat')
xlabel('S')
ylabel('t')
title('European call option, V(S,t)')
```

Using the well known Black-Scholes formula for European call options we can generate the exact solution in figure 1 and calculate the errors of the two methods in figure 4. From the figure we see that the CN method has smaller errors as expected as it is a second order method and the explicit is a first order method.

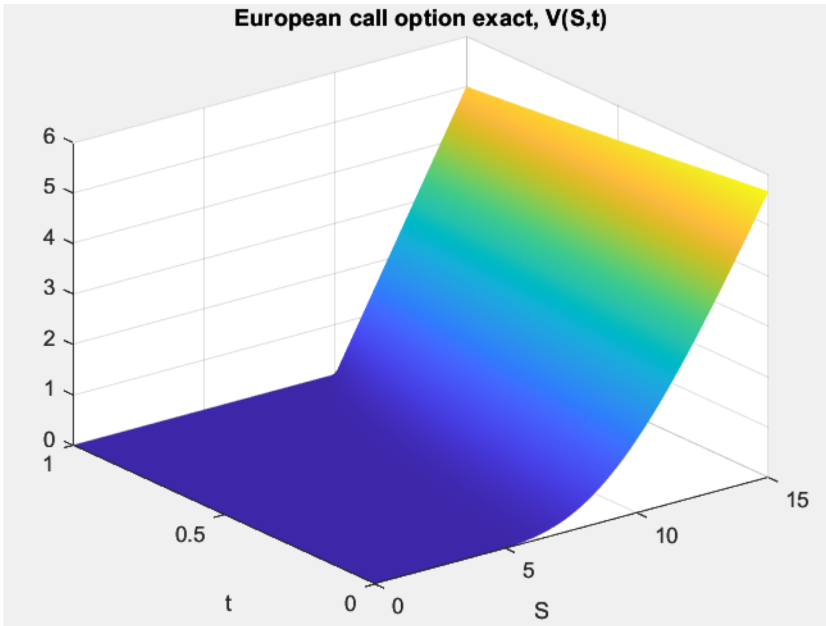
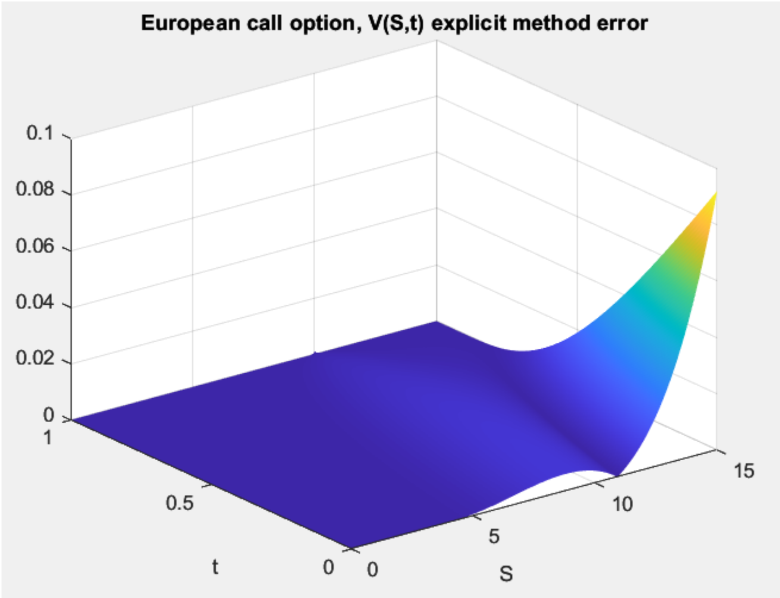
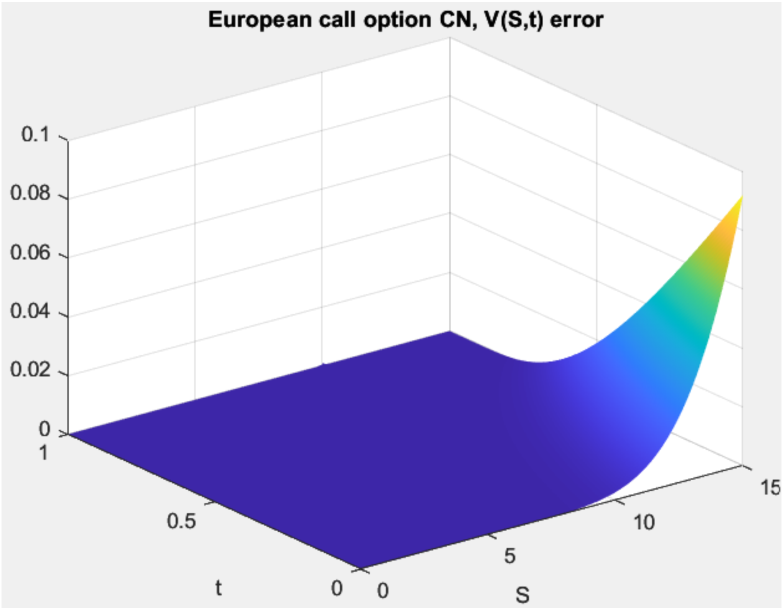


Figure 3: Plot of $V(S,t)$ in the case of a European call option, obtained from the Black-Scholes formula.



(a) Error for the explicit method.



(b) Error for the CN method.

Figure 4: Errors for the two method.

c) Consider the case of a European put option with parameters $r = 0.08$, $\sigma = 0.4$, $T = 4$ and $K = 5$. Imagine you would be asked to calculate the value of the option at time instants $t = 1$, $t = 2$ and $t = 3$ assuming that for all those instants of time the market price of the asset is $S = 5$. Run the routine with different discretization parameters h_t and h_S in order to have an intuition about the error of the approximations for values for $V(5,1)$, $V(5,2)$ and $V(5,3)$ in such a way to show approximations for those values with two decimal digits.

To calculate the different values we simply modified the number of points sampled in the grid and the boundary conditions for the problem in (a) so it now solves the problem for a put option. For the explicit method, the stability conditions limit the range of sample sizes. The code to sample the solution at different points in time is in the functions `explicit.m` and `CN.m`. For the construction of the tables we chose $S_{\text{star}} = 40$.

In the following tables we will take a look at how the error changes with the change of h_S and h_t. We can calculate the exact method for the put option using the well known formula which is programmed in `exact.m` and from there the errors are calculated in `method-comparison.m`. The value of the option at time $t = 1$ is 0.75, at time $t = 2$ it is 0.70 and at time $t = 3$ it is 0.58. From the tables we can see that the CN method converges faster to the true solution since it is second order in time and space, while the explicit method is second order in time and first order in space.

Note: The value of the options decreases as the options is closer to maturity which makes sense.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/20	0.18	0.18	0.18	0.18	0.18
1/40	0.04	0.04	0.04	0.04	0.04
1/80	0.06	0.06	0.06	0.06	0.06
1/160			0.04	0.04	0.04
1/320					0.02

Table 1: Errors calculated for the explicit method at time $t = 1$.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/20	0.25	0.25	0.25	0.25	0.25
1/40	0.03	0.03	0.03	0.03	0.03
1/80	0.01	0.01	0.01	0.01	0.01
1/160	0	0	0	0	0
1/320	0	0	0	0	0

Table 2: Errors calculated for the Crank-Nicolson method at time $t = 1$.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/ 20	0.16	0.16	0.16	0.16	0.16
1/ 40	0.07	0.07	0.07	0.07	0.07
1/ 80	0.05	0.05	0.05	0.05	0.05
1/ 160			0.03	0.03	0.03
1/ 320					0.02

Table 3: Errors calculated for the explicit method at time $t = 2$.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/ 20	0.29	0.29	0.29	0.29	0.29
1/ 40	0.04	0.04	0.04	0.04	0.04
1/ 80	0.02	0.02	0.02	0.02	0.02
1/ 160	0.01	0.01	0.01	0.01	0.01
1/ 320	0	0	0	0	0

Table 4: Errors calculated for the Crank-Nicolson method at time $t = 2$.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/ 20	0.14	0.14	0.14	0.14	0.14
1/ 40	0.07	0.07	0.07	0.07	0.07
1/ 80	0.04	0.04	0.04	0.04	0.04
1/ 160			0.02	0.02	0.02
1/ 320					0.01

Table 5: Errors calculated for the explicit method at time $t = 3$.

h_s/h_t	**1/625**	**1/1250**	**1/2500**	**1/5000**	**1/10000**
1/ 20	0.29	0.29	0.29	0.29	0.29
1/ 40	0.05	0.05	0.05	0.05	0.05
1/ 80	0.02	0.02	0.02	0.02	0.02
1/ 160	0.01	0.01	0.01	0.01	0.01
1/ 320					0.00

Table 6: Errors calculated for the Crank-Nicolson method at time $t = 3$.