COMPSCI 101: Assignment #1

Due on Monday, January 1, 2012

Jones 10:30am

John Smith

Contents

Problem 1

[Summary]

[Statement of the problem]

1.Compute the values of the European vanilla put for $E^* = \$10$, $r^* = 0.05/yr$, $\sigma^* = 0.20/yr$ and with a six month expiry with and without Rannacher smoothing. Report the error as a function of δS and $\delta\tau$. Compute the greeks, δ and Γ and their errors. Propose and implement a technique to compute the $v = \partial V/\partial\sigma$ and report on its performance. Test the effects of the outer boundary on the solution in the range $[0, K]$.

2. Redo the previous task for the European binary put. In particular, examine the solution for large $\delta\tau$ and no smoothing

[Description of The Mathematics]

The BSM and boundary condition for European put option is:

$$\begin{cases} V_t + \frac{\sigma^2 S^2}{2} V_{SS} + rSV_s - rV = 0 \\ V(0,t) = Ke^{-r(T-t)} \\ V(\infty, t) = 0 \\ V(S,T) = max(K - S, 0) \end{cases}$$

combine : $\delta_t^+ V_j^n + L_h[\alpha V_j^{n+1} + (1-\alpha)V_j^n] = 0$
$\alpha = 1$: Back Euler
$\alpha = \frac{1}{2}$: Crank-Nicolson method

Backward Euler method:
The scheme for the Backward Euler method is given by:

$$\frac{V_{i,j} - V_{i,j-1}}{\delta t} + \frac{1}{2}\sigma^2(i\delta S)^2 \frac{V_{i+1,j-1} - 2V_{i,j-1} + V_{i-1,j-1}}{\delta S^2} + r(i\delta S)\frac{V_{i+1,j-1} - V_{i-1}j - 1}{2\delta S)} - rV_{i,j-1} = 0$$

we can rewrite it as:

$$V_{i,j} = A_i V_{i-1,j-1} + B_i V_{i,j-1} + C_i V_{i+1,j-1}$$

where:

Problem 1 continued on next page...

$$A_i = \frac{1}{2}\delta t(r_i - \sigma^2 i^2), B_i = 1 + (\sigma^2 i^2 + r)\delta t, C_i = -\frac{1}{2}\delta t(ri + \sigma^2 i^2)$$

Crank-Nicolson method:

$$\frac{V_{ij} - V_{i,j-1}}{\delta t} + \frac{ri\delta S}{2} + (\frac{V_{i+1,j-1} - V_{i-1,j-1}}{2\delta S} + \frac{ri\delta S}{2}(\frac{V_{i+1,j} - V_{i-1,j}}{2\delta S}) +$$
$$\frac{\sigma^2 i^2 (\delta S)^2}{4}(\frac{V_{i+1,j-1} - 2V_{i,j-1} + V_{i-1,j-1}}{(\delta S)^2}) +$$
$$\frac{\sigma^2 i^2 (\delta S)^2}{4}(\frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{(\delta S)^2}) = \frac{r}{2}V_{i,j-1} + \frac{r}{2}V_{ij}$$

We can rewrite the above equation as:

$$-\alpha_i V_{i-1,j-1} + (1 - \beta_i)V_{i,j-1} - \gamma_i V_{i+1,j-1} = \alpha_i V_{i-1,j} + (1 + \beta_i)V_{i,j} + \gamma_i V_{i+1,j}$$

Where:

$$\alpha_i = \frac{\Delta t}{4}(\sigma^2 i^2 - ri)$$
$$\beta_i = -\frac{\Delta t}{2}(\sigma^2 i^2 + r)$$
$$\gamma_i = \frac{\Delta t}{4}(\sigma^2 i^2 + ri)$$

Ranacher Smooth methods

(1)We use backward Euler for a few n $\geq$ 2 time steps

(2)Use Crank-Nicolson after that: given second order accuracy

Close form Black Scholes formula To test the result for the SDE model of the option pricing, we also need to know the close form solution of the Black -Scholes assumptions, which is the famous Black- Scholes formula.
For the European put options:

$$P(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1)$$

where:

$$d_1 = \frac{log\frac{S}{K} + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$
$$d_2 = \frac{log\frac{S}{K} + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$
$$N(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty} xe^{-\frac{1}{2}S^2}ds$$

Delta for Vanilla put option:

$$-e^{-q\tau}\Phi(-d_1)$$

Gamma for Vanilla put option:

$$-e^{-q\tau}\frac{\Phi(d_1)}{S\sigma\sqrt{\tau}}$$

Vega for Vanilla put option:

$$Se^{-q\tau}\phi(d_1)\sqrt{\tau} = Ke^{-r\tau}\phi(d_2)\sqrt{\tau}$$

where q here is the dividend rate which is equal to 0 in our problem.
Binary put option:
The binary put option is that the payoff will be 1 if the maturity time stock price is lower than the strike price,otherwise it will be 0;
All the other assumptions are same with the vanilla put option.So we can rewrite the formula as follows:

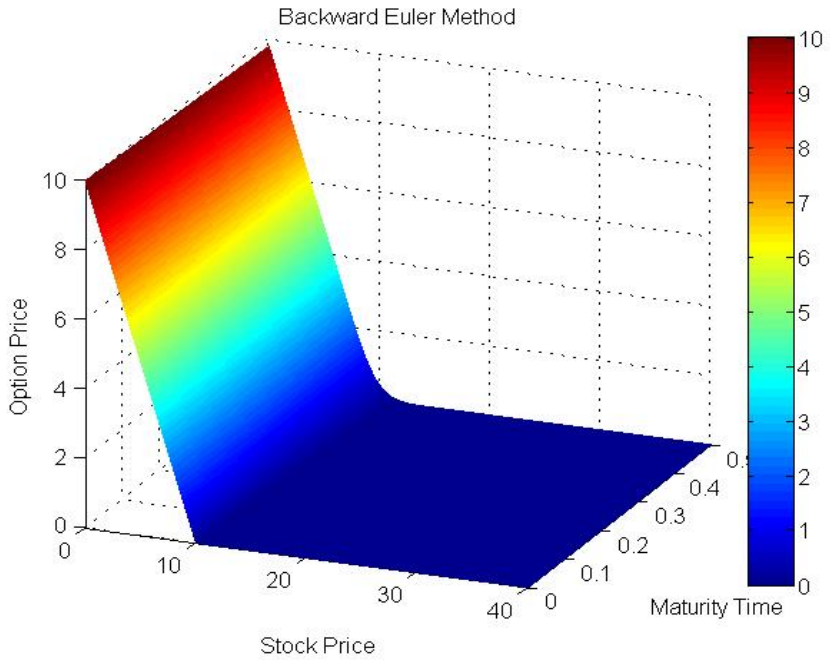
Note that the boundary conditions changed. $\begin{cases} V_t + \frac{\sigma^2 S^2}{2}V_{SS} + rSV_s - rV = 0 \\ V(0,t) = e^{-r(T-t)} \\ V(\infty,t) = 0 \\ V(S,T) = I_{\{S\leq K\}} \end{cases}$

[Results]

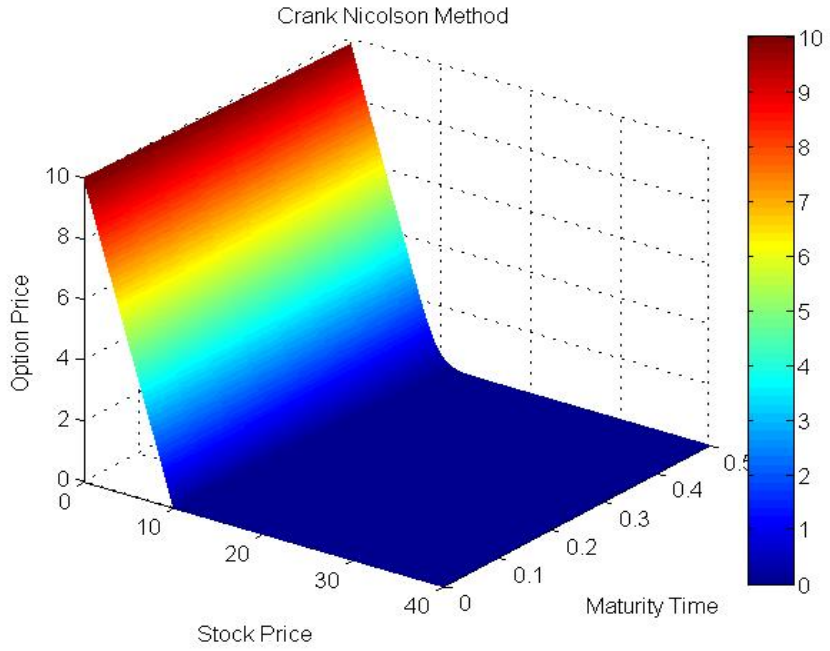
We used $E^* = \$10$, $r^* = 0.05/yr$, $\sigma^* = 0.20/yr$ T=0.5 as the example to build our model.
First, for the regular vanilla put option:

We run the models with different time and stock price steps, and the following is are results when we choose the number of time steps and the number of stock price steps both equal to 640.
Option price

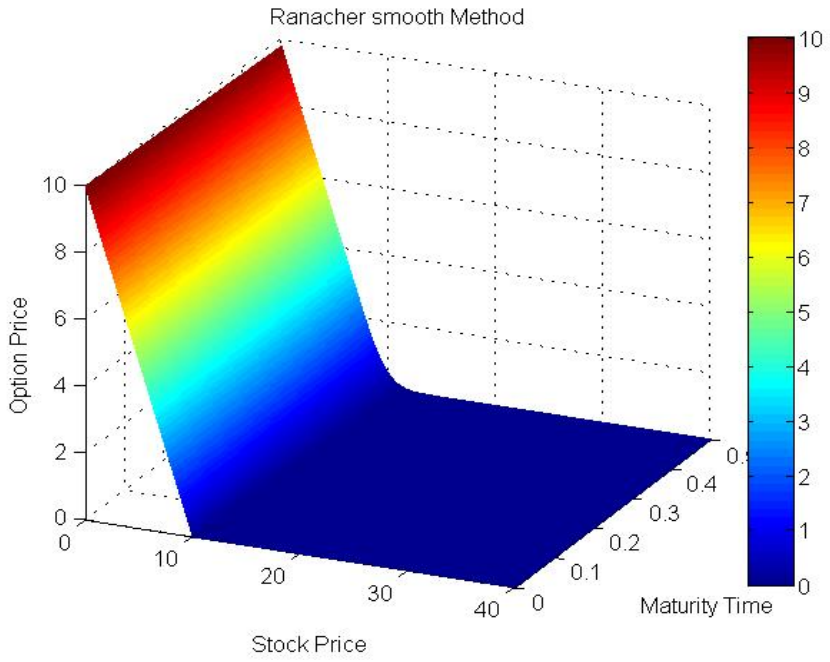
The surface of option price under Backward Euler Methods:



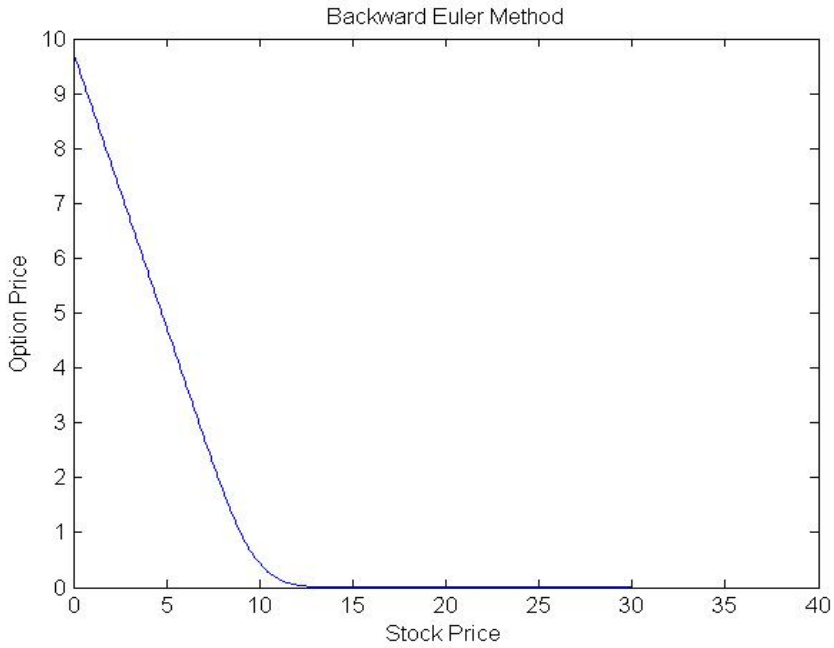
The surface of option price under Crank Nicolson Methods:



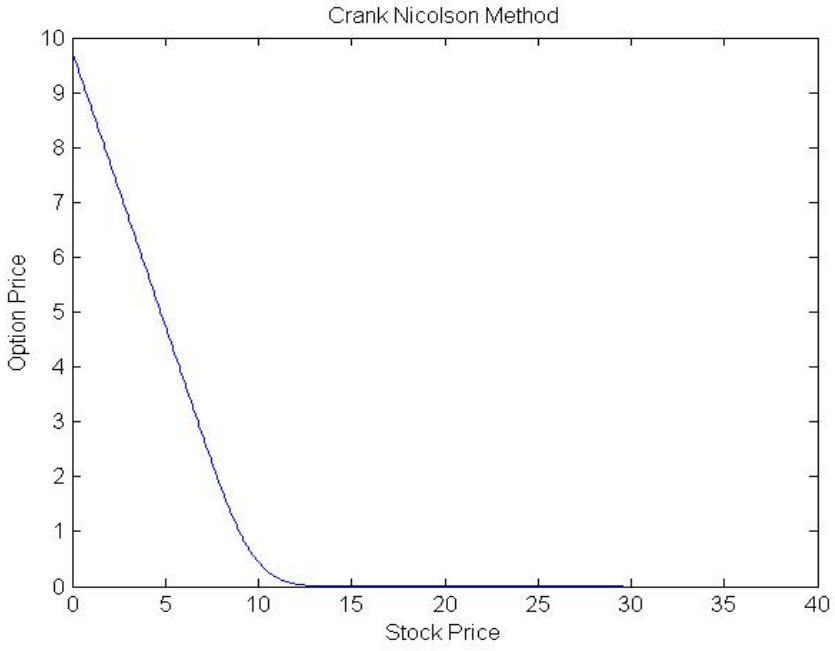
The surface of option price under Ranacher Smooth Methods:



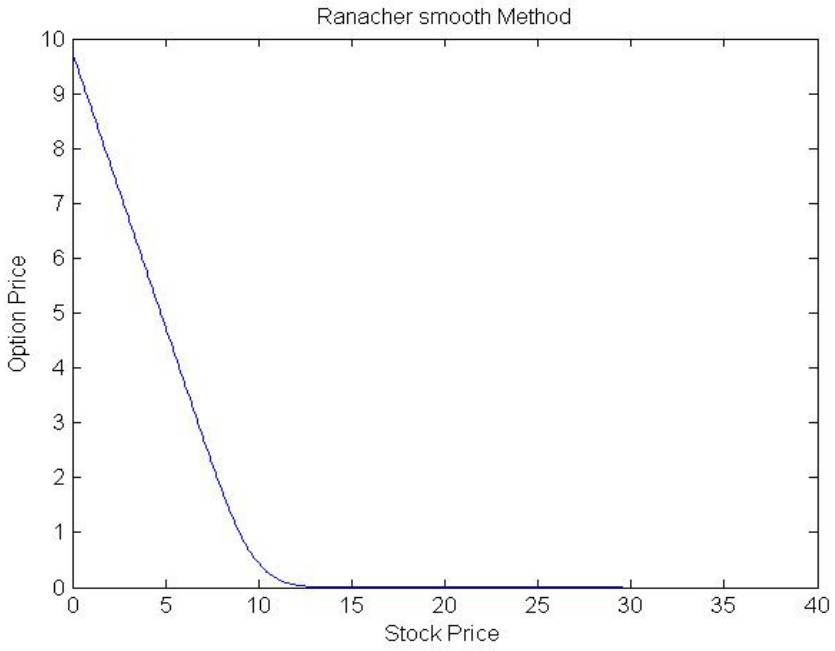
The option price under different stock prices of Backward Euler method:



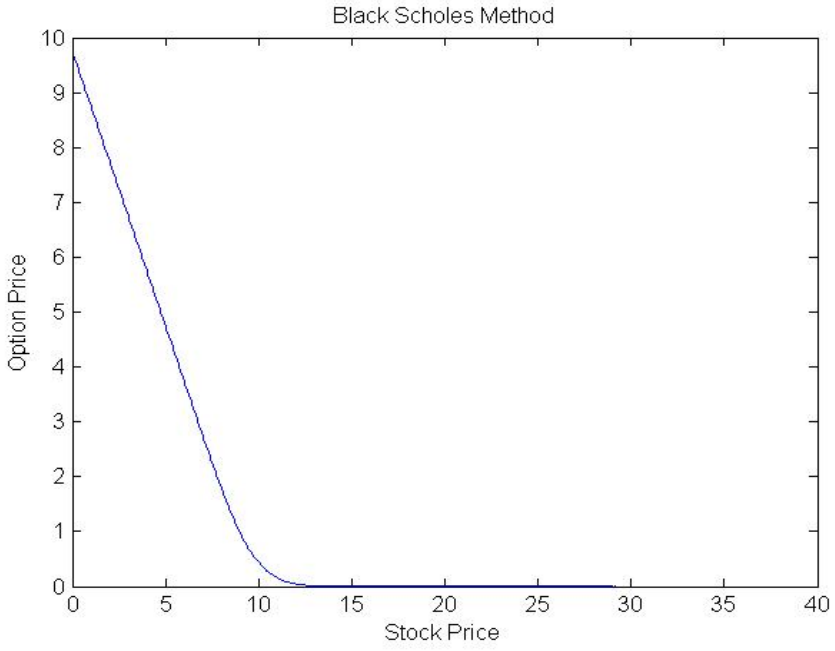
The option price under different stock prices of Crank Nicolson method:



The option price under different stock prices of Ranacher Smooth method:



The option price under different stock prices of Ranacher Smooth method:



Error Test

First set the N_x equal to 640 and see the error change based on the change of N_t

N_t	Backward	Rate	Crank Nicolson	Rate	Ranacher Smooth	Rate
20	0.003594387		0.001567487		0.012230367	
40	0.001867687	1.92	0.000132867	11.80	0.006112979	2.00
80	0.00100197	1.86	0.000135258	0.98	0.003055939	2.00
160	0.000568873	1.76	0.000137234	0.99	0.001527832	2.00
320	0.000352555	1.61	0.000137727	1.00	0.000763881	2.00
640	0.000244465	1.44	0.000137851	1.00	0.000396422	1.93

Next set the N_t equal to 640 and see the error change based on the change of N_x

N_x	Backward	Rate	Crank Nicolson	Rate	Ranacher Smooth	Rate
20	0.163436618		0.163344756		0.163662174	
40	0.040302554	4.06	0.040159448	4.07	0.040454358	4.05
80	0.009109848	4.42	0.008996653	4.46	0.009257477	4.37
160	0.00231537	3.93	0.002206775	4.08	0.00246187	3.76
320	0.00065681	3.53	0.000552129	4.00	0.000802989	3.07
640	0.000244465	2.69	0.000137851	4.01	0.000396422	2.03

Finally change both N_t and N_s

N_x	N_t	Backward	Rate	Crank Nicolson	Rate	Ranacher Smooth	Rate
20	20	0.166238921		0.163327854		0.173703389	
40	40	0.042442619	3.92	0.040143068	4.07	0.044943594	3.86
80	80	0.009901673	4.29	0.008993691	4.46	0.011095262	4.05
160	160	0.002640911	3.75	0.002206118	4.08	0.003229567	3.44
320	320	0.000764333	3.46	0.000552005	4.00	0.001062514	3.04
640	640	0.000244465	3.13	0.000137851	4.00	0.000396422	2.68

We can see that both

the three methods are first order converge on N_t and second order converge on N_x. When both N_t and N_x change, the three methods are second order converge.