E4732 CaseStudy 1 name: Yunfei Yan UNI: yy2516

(1) Description of coding environment.

I use Mobaxterm to code this case study 1 @clic-lab.cs.columbia.edu in linux environment, my compiler is git and my editor is vim.

I haven't used any external library to do the math calculation.

(2) My answers to the question:

S=100, K=90, r=0.0025, q=0.0125, volatility=0.5, tau=1.

Price of Black Scholes Formula:

14.448308

*Price of Finite Difference Methods in **2nd order** approximation for **delta, gamma** and **Neumann Boundary Condition** by tridiagonal matrix solver:*

14.438311

*Price of Finite Difference Methods in **3rd order** approximation for **gamma** and **Neumann Boundary** and **4th order** approximation for **delta** by pentadiagonal matrix solver:*

14.438298

Some induction and results about pentadiagonal matrix solver:

$$f^{(2)}(x)=\frac{16f(x+h)+16f(x-h)-30f(x)-f(x+2h)-f(x-2h)}{12h^2}+O(h^3)$$

$$f^{(1)}(x)=\frac{8f(x+h)-8f(x-h)-f(x+2h)+f(x-2h)}{12h}+O(h^4)$$

Let: $\begin{aligned}&\frac{\partial v^2}{\partial S^2}(S\min+2\Delta S,\tau_{k+1})=0\\&\frac{\partial v^2}{\partial S^2}(S\max-2\Delta S,\tau_{k+1})=0\end{aligned}$ *, and use central approximation.*

We yield the first pair Neumann Boundary Condition:

$$\begin{aligned}v_{3,k}&=(m_{3,k+1}-k_{3,k+1})v_{5,k+1}+(16k_{3,k+1}+u_{3,k+1})v_{4,k+1}+\\&(d_{3,k+1}-30k_{3,k+1})v_{3,k+1}+(l_{3,k+1}+16k_{3,k+1})v_{2,k+1}\\v_{N-1,k}&=(k_{N-1,k+1}-m_{N-1,k+1})v_{N-3,k+1}+(l_{N-1,k+1}+16m_{N-1,k+1})v_{N-2,k+1}+\\&(d_{N-1,k+1}-30m_{N-1,k+1})v_{N-1,k+1}+(u_{N-1,k+1}+16m_{N-1,k+1})v_{N,k+1}\end{aligned}$$

Then, we let: $\begin{aligned}&\frac{\partial v^2}{\partial S^2}(S\min+\Delta S,\tau_{k+1})=0\\&\frac{\partial v^2}{\partial S^2}(S\max-\Delta S,\tau_{k+1})=0\end{aligned}$ *, and use forward approximation:*

$$f^{(2)}(x)=\frac{6f(x+h)+11f(x-h)-f(x+3h)+4f(x+2h)-20f(x)}{12h^2}+O(h^3)$$

We yield the second pair Neumann Boundary Condition:

$$\begin{aligned}6v_4+11v_1-v_5+4v_4-20v_2&=0\\16v_4+16v_2-30v_3-v_5-v_1&=0\end{aligned}$$

, remove v_1 and replace v_2 in the first pair Neumann Boundary Condition, we have:

$$v_2=\frac{27v_3+v_5-15v_4}{13}$$

$$v_{3,k}=(m_{3,k+1}+\frac{1}{13}l_{3,\mathrm{k}+1}+\frac{3}{13}k_{3,k+1})v_{5,k+1}$$

$$+(u_{3,k+1}-\frac{15}{13}l_{3,\mathrm{k}+1}-\frac{32}{13}k_{3,k+1})v_{4,k+1}$$

$$+(d_{3,k+1}+\frac{27}{13}l_{3,\mathrm{k}+1}+\frac{42}{13}k_{3,k+1})v_{3,k+1}$$

Similarly

$$\begin{aligned}6v_{N-1}+11v_{N+1}-v_{N-3}+4v_{N-2}-20v_N&=0\\16v_N+16v_{N-2}-30v_{N-1}-v_{N-3}-v_{N+1}&=0\end{aligned}$$

,remove v_{N+1}, and replace v_N in the first pair Neumann Boundary Condition, we have:

$$v_N=\frac{27v_{N-1}+v_{N+3}-15v_{N-2}}{13}$$

$$v_{N-1,k}=(k_{N-1,k+1}+\frac{1}{13}u_{N-1,k+1}+\frac{3}{13}m_{N-1,k+1})v_{N-3,k+1}$$

$$+(l_{N-1,k+1}-\frac{15}{13}u_{N-1,\mathrm{k}+1}-\frac{32}{13}m_{N-1,k+1})v_{N-2,k+1}$$

$$+(d_{N-1,k+1}+\frac{27}{13}u_{N-1,k+1}+\frac{42}{13}m_{N-1,k+1})v_{N-1,k+1}$$

Finally, we can replace the v_2 in the difference equation when $j=4$, and replace the v_N in the difference equation when $j=N-2$, and get the following result:

$$v_{4,k}=m_{4,k+1}v_{6,k+1}+(\frac{1}{13}k_{4,k+1}+u_{4,k+1})v_{5,k+1}$$

$$+(d_{4,k+1}-\frac{15}{13}k_{4,k+1})v_{4,k+1}+(l_{4,\mathrm{k}+1}+\frac{27}{13}k_{4,k+1})v_{3,k+1}$$

$$v_{N-2,k}=k_{N-2,k+1}v_{N-4,k+1}+(l_{N-2,k+1}+\frac{1}{13}m_{N-2,k+1})v_{N-3,k+1}$$

$$+(d_{N-2,k+1}-\frac{15}{13}m_{N-2,k+1})v_{N-2,k+1}+(u_{N-2,k+1}+\frac{27}{13}m_{N-2,k+1})v_{N-1,k+1}$$

(3) Compare and Conclusion:

a.

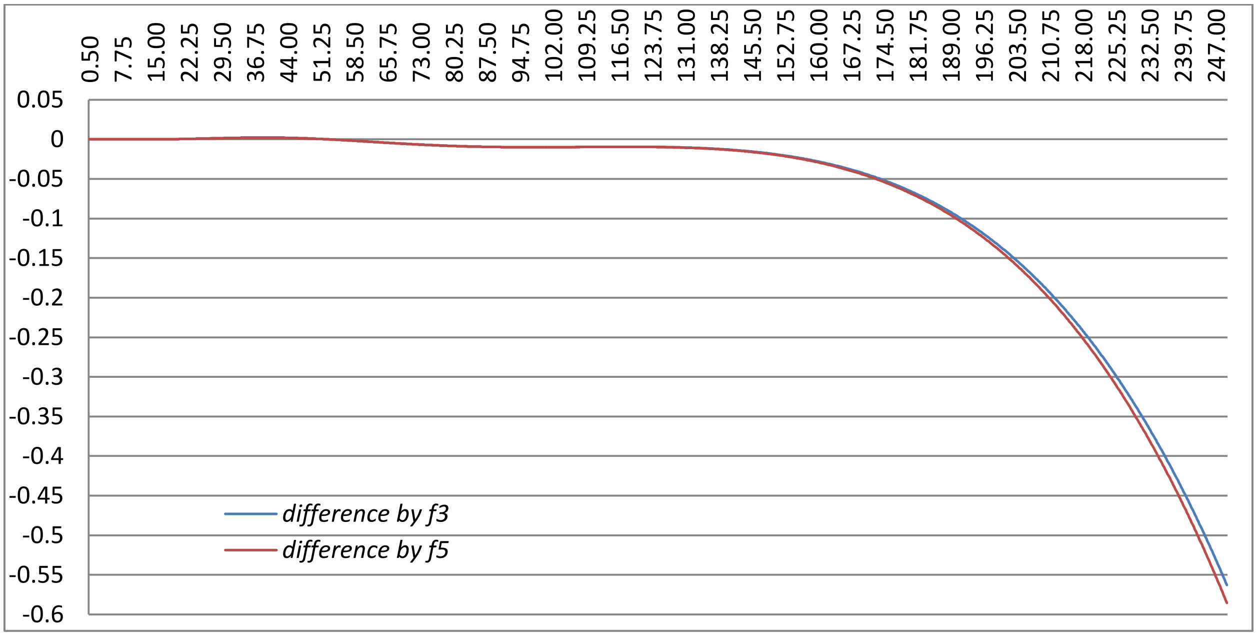
Theoretically, the speed of f3 should be faster than the f5. Due to the machine that I used is the one in Columbia clic-lab, which is very powerful, I didn't feel the big difference about speed between these two methods.

b.

Some personal definition about solution:

difference: *prices acquired by finite difference methods - prices acquired by Black-Scholes formula*

I have calculated the Black-Scholes price for all the initial S on my grid and plot the difference both for F3 and F5 ***in the attached Excel,*** *and I put the plot here:*



We can discover several points from the graph:

(i) Both methods deviate from the Black-Scholes formula price heavily with the increase of the initial price of underlying S.

(ii) Both methods are almost exactly the same with the Black-Scholes price, however, there exists a little "wave" phenomenon ***which can be ignored*** *at the beginning.*

(iii) Generally, the curve of f5 is above the curve of f3, which means that the f5 is more consistent with the Black Scholes price in a global version and the accuracy of finite difference method decreases as the S approaches the Smax area.

(iiii) The prices yielded from finite difference method would be lower than the Black-Scholes prices in a global vision.

(4) Description of the logic present in the written source code

In f3.cc, I have written the ***main function*** *and* ***tridiagonalsolver function****. I initialed all the variables in the main and called the tridiagonalsolver in the main body. (a little difference with others is that I have already modified the d array before I passed the l array, d array, u array and v array into the tridiagonalsolver which would improve the speed of calculation.)*

In f5.cc, I have written the ***main function*** *and* ***pentadiagonalsolver function****. I initialed all the variables in the main function and called pentadiagonalsolver in the main body. In pentadiagonalsolver, I use malloc for all k, l, d, u, m, and v which would not change the value of these origin arrays.*

I download my code from Mobaxterm directly with .cc type. You can open it using notepad (I have already tried, and it works.) If you cannot open it, please contact me.

Thank you!