End-user's instruction

January 2022

1 Introduction

This is an end-user guidance document. It is intended to guide the end user to the correct implementation of this implementation, select functions, configure parameters, and interpret the displayed results.

2 What contain in the zip file

When you unzip the file you will find a number of files which are MATLAB scripts, two documents, the developer documentation and end-user's instruction (this document). The MATLAB scripts (.m files) consist of the following files:

1. BarrierCallOption_BS.m
2. CrankNicolsonFDM.m
3. tridiag.m
4. tridiag_prod.m
5. test.m
6. test_impact_of_M.m
7. test_impact_of_N.m

3 A brief description of the methods implemented.

3.1 the Black-Scholes

Given the interest rate r, the dividend yield d and volatility σ. Assume that the stock price S follows the process

$$dS_t = (r - d)S_t dt + \sigma S_t dW_t,$$

where $W = (W(t) : t \geq 0)$ is a Wiener process under the risk-neutral probability measure.

Let $V(S,t)$ be the option price on the above stock satisfies the Black-Scholes (BS) equation:

$$\frac{\partial V}{\partial t}+\frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2+(r-d)S\frac{\partial V}{\partial S}=rV,$$

where $-\infty \leq S \leq \infty$ and $0 \leq t \leq T$. And with final condition $V(S,T)=f(T)$.

One can use the change of variables below to simplify BS equation;

$$S=\exp{(x)}, t=T-\frac{2\tau}{\sigma^2}, V(S,t)=v(x,\tau)=v\left(\ln(x),\frac{\sigma^2}{2}(T-t)\right).$$

We, thus, have

$$\frac{\partial v}{\partial \tau}=\frac{\partial^2 v}{\partial x^2}+\left(\frac{2(r-d)}{\sigma^2}-1\right)\frac{\partial v}{\partial x}-\frac{2r}{\sigma^2}v,$$

with $v(x,0)=V(\exp(x),T)=f(\exp(x))$, where $-\infty \leq x \leq \infty$ and $0 \leq \tau \leq \frac{\sigma^2}{2}T$.

Putting that $v(x,\tau)=\exp(\alpha x+\beta\tau)u(x,\tau)$, when

$$\alpha=\frac{\sigma^2-2r+2d}{2\sigma^2}, \beta=-\left(\frac{\sigma^2+2r-2d}{2\sigma^2}\right)^2-\left(\frac{2d}{\sigma^2}\right).$$

Hence we obtain

$$\frac{\partial u}{\partial \tau}=\frac{\partial^2 u}{\partial x^2},$$

with $u(x,0)=\exp(-\alpha x)v(x,0)=\exp(-\alpha x)f(\exp(x))$ when $-\infty \leq x \leq \infty$ and $0 \leq \tau \leq \frac{\sigma^2}{2}T$.

3.2 the Barrier options

In our implementation, we use an up-and-out call option with zero rebate which has a payoff;

$$f(S_T)=\begin{cases}\max(0,S_T-K) & \text{if } \max\{S_t: 0\leq t\leq T\}<B,\\ 0 & \text{otherwise,}\end{cases}$$

where K is a strike(or exercise) price and B is a predetermined (barrier) level. Therefore; the initial condition becomes,

$$u(x,0)=\exp(-\alpha x)\max(0,\exp(x)-K),$$

if $\max\{S_t: 0\leq t\leq T\}<B$ and $u(x,0)=0$ otherwise. Moreover, the boundary conditions for an up-and-out call option are

$$u(x,\tau)=\exp(-\alpha x-\beta\tau)v(x,\tau)=\exp(-\alpha x-\beta\tau)V(0,\tau)=0 \text{ as } x\to-\infty$$

and

$$u(x=ln(B),\tau)=\exp(-\alpha x-\beta\tau)V(B,\tau)=0.$$

So it is zero boundary condition.

3.3 the Finite Difference Methods (FDM)

Our implementation used the Crank-Nicolson finite difference method for solving the barrier call option. It is a FDM which is a combination of the explicit and implicit schemes for FDM. It solve the heat equation below;

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2},$$

for $a \leq x \leq b$ and $0 \leq t \leq T$, where $\kappa > 0$. With the initial condition; for $a \leq x \leq b$,

$$u(x, 0) = f(x)$$

and the boundary condition; for $0 \leq t \leq T$,

$$u(a, t) = g(t), u(b, t) = h(t).$$

Select N grid points in the space interval $[a, b]$ which has $\Delta x = \frac{b-a}{N}$ and select M grid points in the time interval $[0, T]$ which has $\Delta t = \frac{T}{M}$. Let U_k be the approximate PDE solution for the time slice t_k. The recursive explicit discretization scheme in matrix form for the explicit and implicit schemes are

$$U_{k+1} = A_E U_k + B_k$$

and

$$A_I U_{k+1} = U_k + B_{k+1}.$$

Therefore; for the Crank-Nicolson scheme, we have

$$U_{k+1} = (I + A_I)^{-1}(I + A_E)U_k + (I + A_I)^{-1}(B_k + B_{k+1}),$$

for $1 \leq k \leq M$.

Note: A_E, A_I and B can be constructed by using the value of ρ, $h(t)$ and $g(t)$ And $\rho = \kappa \frac{\Delta t}{\Delta x^2}$.

Hence, by performing this scheme, we will obtain the approximation solution U for the heat equation.

4 Example

The following is an example of using my implementation for calculating barrier option prices with the Crank-Nicolson FDM (as in the test.m file).

```
%Clear memory and console output
clc
clear

%Set the number of grid points
N = 500;          % For the space interval [a,b]
M = 100;          % For the time interval [0,T]
```

```
%Paramerters for Barrier Option Pricing
S0 = 95;         % Stock Price
K = 110;         % Strike Price
B = 4*K;         % Barrier level (use 4*K)
Smin = 0.001;    % Set minimum stock price
Smax = 4*K;      % Set maximum stock price
T = 1;           % Term (years)
r = 5;           % Interest Rate (%)
d = 0;           % Dividend Yield (%)
sigma = 25;      % Volatility (%)

%Solve FDM for BS
[S, t, V] = BarrierCallOption_BS(Smin,Smax,T,N,M,r,d,
    sigma,K,B,true);

%Calculate Up-and-out Barrier option price when the Stock
     Price is S0
interp1(S,V(:,end),S0)
```

To use *BarrierCallOption_BS* function for find the call option price, we need to set the input for the function, which are $(Smin, Smax, T, N, M, r, d, sigma, K, B, plt)$. These inputs consist of the minimun stock price (Smin), the maximum stock price (Smax), strike price (K), interest rate (r, in %), dividend yield rate (d, in %), volatility rate (sigma, in %), barrier level (B), the number of grid points for the space (N) and time intervals (M). And the boolean parameter (plt) controls whether the plots, the prices of the option at time 0 against the stock spot prices and the surface plot for the option prices corresponding to the FDM grid in S and t, will be displayed.

This example takes values for each parameters as its shown in the code, one will obtains the plot for the option at time 0 against the stock spot price and the 3D plot for the option prices with the whole FDM grid in S and t as in the figures (1, 2).

Moreover, one can obtain the call option price (corresponding to the given the Stock Price $S0$) by using the output (S,V) of *BarrierCallOption_BS* and the function *interp1* as it shown above, and it is 5.8424.

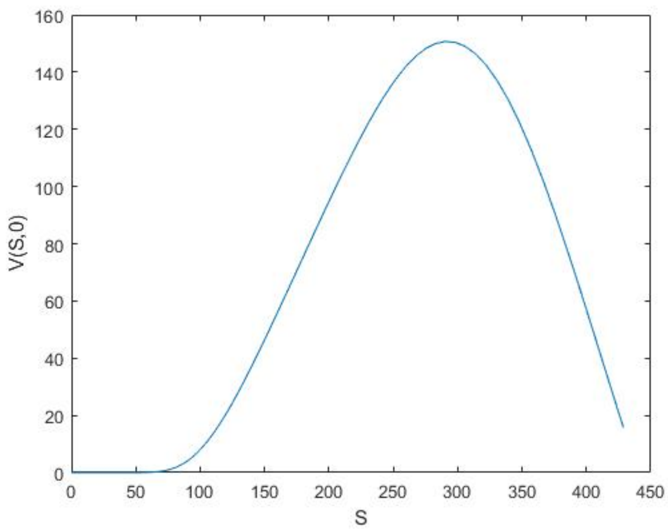

Figure 1: the option at time 0 plot against the stock spot price

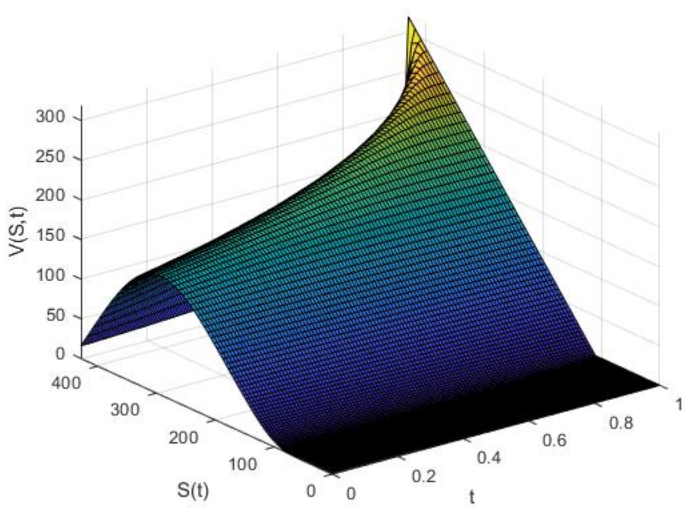

Figure 2: the option prices plot with the whole FDM grid in $S(t)$ and t