End-User Instructions

In this document a short description on how to use the Explicit.m, Implict.m, Crank.m and monte_carlo.m codes is going to be made. Given that the structure of all of the codes mentioned above is very similar the instructions to use those code is going to be the same.

First of all for using the codes above the user must enter some parameters directly into the section "Parameters" of the script. The parameters requiered are:

- K: Strike Price
- S_o: Spot Price
- r: Interest rate
- q: Dividend yield
- B: Barrier Level
- T: Time to maturity
- α: Exponent in local volatibity function
- N: Total number of divisions on the price domain
- M: Total number of time steps desired.
- $NSim$: Total number of paths generated. (Only requiered for monte_carlo.sim)

Once having entered the parameters, the user must run the code. All of the codes are going to solve the Black Scholes PDE (Equation 1))with the following initial and boundary conditions.

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 + (r-q)S\frac{\partial V}{\partial S} - rV \tag{1}$$

Initial Condition:

$$V(S,0) = \begin{cases} max(S-K,0) & S < B \\ 0 & Otherwise \end{cases} \tag{2}$$

Boundary Conditions:

$$V(0,\tau) = 0 \qquad\qquad V(B,\tau) = 0 \tag{3}$$

Once the code have finish running, they are going to report the results of the PDE. For the FDM codes these are going to print the value of $V(S_o, \tau = T)$ as well as two figures. The first figure corresponds to the profile $V(S, \tau = T)$ vs S for all values of S. The second figure corresponds to a 3D surface which shows $V(S,\tau)$ vs S vs τ. Regarding the results reported by the monte caro code, this only prints the value of $V(S_o, \tau = T)$ in console and genrate one figure which corresponds to the profile $V(S, \tau = T)$ vs S.

Description of the numerical Methods implementation

The implementation of all the numerical methods in the codes is going to be explained in this section.

For discretizing the right hand side of equation 1, second order central difference schemes were used. This discretization of the "spatial" (are no spatial in an strick manner but derivate with respect to S are going to be refered in that way thorughout the document) derivatives on equation 1 is shown in the following equations.

$$\frac{1}{2}\frac{\partial^2 V}{\partial S^2}\sigma^2 S^2 \approx \frac{1}{2}\frac{V_{j+1}-2V_j+V_{j-1}}{\Delta S^2}\sigma^2(\tau,S_j)S_j^2 \tag{4}$$

$$(r-q)S\frac{\partial V}{\partial S} \approx (r-q)S_j\frac{V_{j+1}-V_{j-1}}{2\Delta S} \tag{5}$$

$$rV \approx rV_j \tag{6}$$

With the previous discretization defined The right hand side of equation 1 can be written as

$$F[V^n,S_j,\tau^n] = \frac{1}{2}\frac{V_{j+1}-2V_j+V_{j-1}}{\Delta S^2}\sigma^2(\tau,S_j)S_j^2 + (r-q)S_j\frac{V_{j+1}-V_{j-1}}{2\Delta S} - rV_j \tag{7}$$

Now with respect to the initial conditions for all cases this was calculated using the up and out call option which written in discretized way would be

$$V_j^0 = \begin{cases} max(S_j-K,0) & S_j < B \\ 0 & Otherwise \end{cases} \quad \forall j, 1<j<M \tag{8}$$

On the other hand, for all cases the boundary conditions would be expressed in discretized form as

$$V_0^n = 0 \qquad\qquad V_M^n = 0 \forall n, 0<n<N \tag{9}$$

Where N is the total number of time steps taken and M is the total division on the spatial domain.

Explicit Finite Difference Method.

For the Explicit finite difference method a first order foward Euler scheme was used which can be written as:

$$\frac{V_j^{n+1}-V_j^n}{\Delta\tau} = F[V^n,S,\tau^n] \tag{10}$$

Replacing equation 7 in the previous expression and solving for V_j^{n+1} we get

$$
\begin{aligned}
V_j^{n+1} = &\left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^n, S_j)S_j^2 + (r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j+1}^n \\
&+ \left(-\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^n, S_j)S_j^2 - r\Delta\tau V_j + 1\right)V_j^n \\
&+ \left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^n, S_j)S_j^2 - (r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j-1}^n
\end{aligned} \tag{11}
$$

Tre previous equation is solve directly for each time step.

Implict Finite Difference Method.

For the Implict finite difference method a first order Backward Euler scheme was used which can be written as:

$$
\frac{V_j^{n+1} - V_j^n}{\Delta\tau} = F[V^{n+1}, S, \tau^{n+1}] \tag{12}
$$

Replacing equation 7 in the previous expression and solving for V_j^n we get

$$
\begin{aligned}
V_j^n = &-\left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1}, S_j)S_j^2 + (r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j+1}^n \\
&- \left(-\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1}, S_j)S_j^2 - r\Delta\tau V_j - 1\right)V_j^{n+1} \\
&- \left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1}, S_j)S_j^2 - (r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j-1}^{n+1}
\end{aligned} \tag{13}
$$

The previous equation is solve by the method of backward substitution given that a system of linear equations is needed to be solve for every time step taken.

Crank-Nicholson Finite Difference Method.

For the Crank-Nicholson difference method finite difference method the following scheme was used:

$$
\frac{V_j^{n+1} - V_j^n}{\Delta\tau} = \frac{1}{2}\left([V^n, S, \tau^{n+1} + F[V^{n+1}, S, \tau^n]\right) \tag{14}
$$

Replacing equation 7 in the previous expression and rearrenging terms so that anythin dependt on the time τ^n is on one side of the equation and everything dependent on τ^{n+1} is in the other side we obtain:

$$
\begin{aligned}
&-\left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2+(r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j+1}^{n}\\
&+\left(2+\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2+r\Delta\tau V_j\right)V_j^{n+1}\\
&+\left(-\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2+(r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j-1}^{n+1}= \quad \left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2+(r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_j^{n}\\
&\qquad +\left(2-\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2-r\Delta\tau V_j\right)V_j^{n}\\
&\qquad +\left(\frac{1}{2}\frac{\Delta\tau}{\Delta S^2}\sigma^2(\tau^{n+1},S_j)S_j^2-(r-q)S_j\frac{\Delta\tau}{2\Delta S}\right)V_{j-1}^{n}
\end{aligned}
\tag{15}
$$

The previous equation is solve by the method of backward substitution given that a system of linear equations is needed to be solve for every time step taken.