

# **Numerical Estimation of Black-Scholes Option Prices using the Explicit Euler Method of Finite Differences**

by

**Hammond Mason**

# Table of Contents

1. Introduction.....	3
2. Deriving the Black-Scholes PDE.....	3
3. Approximating Partial Derivatives using Finite Differences.....	5
3.1 Theta.....	6
3.2 Delta.....	7
3.3 Gamma.....	9
4. The Explicit Euler Method.....	9
4.1 An Algorithm for Explicit Euler.....	11
4.2 The Stability of Explicit Euler.....	17
4.3 Example VBA Code.....	19
4.4 Example Numerical Results.....	20
4.4.1 Option Prices.....	21
4.4.2 Delta and Gamma.....	23
Appendix A: Explicit Euler VBA Code.....	25
Appendix B: Deriving Delta and Gamma.....	27

## 1. Introduction

Finite differences are a tool used in the physical sciences to find numerical solutions to partial differential equations (“PDEs”) for which there is no closed-form/analytical solution. They were first applied to option pricing in the late 1970s when large scale computer power became available to academic institutions. Their continued popularity is due to the inherent flexibility in their application: they can be used to solve (numerically) a wide variety of PDEs.

This paper demonstrates one method of finite differencing – known as the Explicit Euler method – to solve (numerically) the Black-Scholes PDE. Firstly, this partial differential equation is derived which is then followed by a discussion of finite differencing. Secondly, the Explicit Euler method is explained and an algorithm is generated that is coded into Excel VBA. Some numerical results are then discussed.

## 2. Deriving the Black-Scholes PDE

Before we begin to derive an option's price we must make a crucial assumption: we must specify the price process taken by the underlying asset. Black-Scholes makes the assumption that a stock (the underlying asset) follows a standard Geometric Brownian Motion<sup>1</sup> process:

$$\frac{dS}{S} = \mu \cdot dt + \sigma \cdot \sqrt{dt} \cdot dZ$$

ie:

$$dS = S \cdot \mu \cdot dt + S \cdot \sigma \cdot \sqrt{dt} \cdot dZ \dots\dots\dots(2.1)$$

Now consider an option,  $V$ , written on the stock,  $S$ , that pays no cash return (ie. dividends) during the option's term. For each option written (ie. sold) we hold a certain quantity,  $\Delta$ , of the underlying stock. Call this combination of option written (ie. sold) + stock held, the Hedge Portfolio,  $\Pi$ . The value of the Hedge Portfolio at any point in time is, therefore, determined by the formula:

$$\Pi = V - \Delta \cdot S \dots\dots\dots(2.2)$$

Hence, small<sup>2</sup> changes in the Hedge Portfolio, ie.  $d\Pi$ , can be written as:

$$d\Pi = dV - \Delta \cdot dS \dots\dots\dots(2.3)$$

Now, we know that the value of an option is a function of the value of the underlying ( $S$ ) and time ( $t$ ), ie.  $V(S, t)$ . Since  $S$  is stochastic we use Itô's Lemma to arrive at the expression for  $dV$ :

$$dV = \frac{\partial V}{\partial S} \cdot dS + \frac{\partial V}{\partial t} \cdot dt + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot (dS)^2$$

Since we already know  $dS$  (from equation 2.1 above) we can substitute it into this equation to get:

---

<sup>1</sup> Also called a 'Weiner' or 'square-root-of-time' process.  
<sup>2</sup> 'Small' as in infinitesimally small. I shall adhere to the standard convention of using prefix  $d$  to refer to infinitesimally small changes while  $\Delta$  refers to a comparatively, much larger change.

$$dV = \frac{\partial V}{\partial S} \cdot (S \cdot \mu \cdot dt + S \cdot \sigma \cdot \sqrt{dt} \cdot dZ) + \frac{\partial V}{\partial t} \cdot dt + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot (S \cdot \mu \cdot dt + S \cdot \sigma \cdot \sqrt{dt} \cdot dZ)^2$$

Separating out the first term on the RHS and expanding the last term (in the process, recognising through the rules of Itô Calculus that  $dt \cdot dt = 0$ ,  $dt \cdot dZ = 0$  and  $dZ \cdot dZ = \text{var}(dZ) = 1$ ) we arrive at:

$$dV = \frac{\partial V}{\partial S} \cdot (S \cdot \mu \cdot dt) + \frac{\partial V}{\partial S} \cdot (S \cdot \sigma \cdot \sqrt{dt} \cdot dZ) + \frac{\partial V}{\partial t} \cdot dt + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 \cdot dt$$

Grouping the  $dt$  and  $dZ$  terms gives:

$$dV = \left[ \frac{\partial V}{\partial S} \cdot S \cdot \mu + \frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 \right] \cdot dt + \left[ \frac{\partial V}{\partial S} \cdot S \cdot \sigma \cdot \sqrt{dt} \right] \cdot dZ$$

It is worth noting at this point that the second term on the RHS (ie. the  $dZ$  term) is the only source of uncertainty in the value of  $dV$ . The first term (the  $dt$  term) is just a function of the change in time. Substituting this equation for  $dV$  and our equation for  $dS$  into equation 2.3 we get:

$$d\Pi = \left( \left[ \frac{\partial V}{\partial S} \cdot S \cdot \mu + \frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 \right] \cdot dt + \left[ \frac{\partial V}{\partial S} \cdot S \cdot \sigma \cdot \sqrt{dt} \right] \cdot dZ \right) - \Delta \cdot (S \cdot \mu \cdot dt + S \cdot \sigma \cdot \sqrt{dt} \cdot dZ)$$

Grouping the  $dt$  and  $dZ$  terms separately:

$$d\Pi = \left[ \frac{\partial V}{\partial S} \cdot S \cdot \mu + \frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 - \Delta \cdot S \cdot \mu \right] \cdot dt + \left[ \frac{\partial V}{\partial S} \cdot S \cdot \sigma \cdot \sqrt{dt} \right] \cdot dZ - [\Delta \cdot S \cdot \sigma \cdot \sqrt{dt}] \cdot dZ$$

Now, to eliminate the  $dZ$  terms – and thereby remove the sources of uncertainty in  $d\Pi$  – we choose a value of  $\Delta$  such that the last two terms on the RHS equate, ie. we set:

$$\Delta = \frac{\partial V}{\partial S}$$

Then the equation simplifies to:

$$d\Pi = \left[ \frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 \right] \cdot dt \dots\dots\dots(2.4)$$

However, now that there is no uncertainty in  $d\Pi$ , in order for there to be no arbitrage,  $d\Pi$  must change at the risk-free rate<sup>3</sup>, ie:

$$d\Pi = r \cdot \Pi \cdot dt$$

but we know from equation 1.2 the formula for  $\Pi$  so this becomes:

$$d\Pi = r \cdot (V - \Delta S) \cdot dt$$

---

3 The astute reader will have noticed that the  $\mu$  term carried over from equation 1.1 has disappeared. The significance of this cannot be overstated: the price of an option does not depend upon what an investor considers the value of  $\mu$  to be.

expanding the RHS and substituting in our value for  $\Delta$  gives:

$$d\Pi = \left[ r \cdot V - r \cdot \frac{\partial V}{\partial S} \cdot S \right] \cdot dt$$

Substituting equation 2.4 for the LHS:

$$\left[ \frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 \right] \cdot dt = \left[ r \cdot V - r \cdot \frac{\partial V}{\partial S} \cdot S \right] \cdot dt$$

Dividing both sides by  $dt$ :

$$\frac{\partial V}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 = r \cdot V - r \cdot \frac{\partial V}{\partial S} \cdot S$$

which simplifies to:

$$\frac{\partial V}{\partial t} + r \cdot \frac{\partial V}{\partial S} \cdot S + \frac{1}{2} \cdot \frac{\partial^2 V}{\partial S^2} \cdot S^2 \cdot \sigma^2 - r \cdot V = 0$$

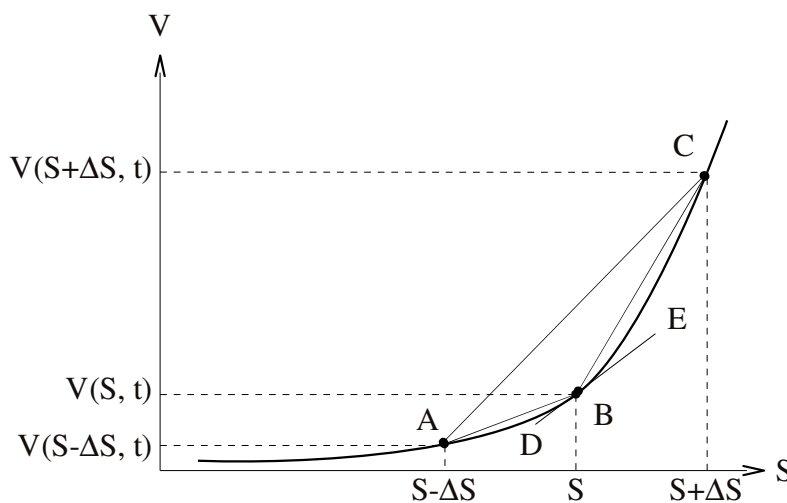
ie:

$$\frac{\partial V}{\partial t} + r \cdot S \cdot \frac{\partial V}{\partial S} + \frac{S^2 \cdot \sigma^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} - r \cdot V = 0 \dots\dots\dots(2.5)$$

which is the Black-Scholes partial differential equation (“PDE”).

### 3. Approximating Partial Derivatives using Finite Differences

Consider the following diagram which shows  $V$  at various levels of  $S$  for a given  $t$  ie. we have frozen time at time= $t$  and are just looking at how  $V$  varies with  $S$ :



Let DE be the tangent to the curve ABC at point B. By definition, the slope of DE is the first derivative of V with respect to S, ie:

$$\text{Slope of DE} = \frac{dV}{dS}$$

It is this value that we wish to estimate using finite differences and we have three estimators we can choose from:

$$\text{grad AB} = \frac{V(S, t) - V(S - \Delta S, t)}{\Delta S} = \text{Backward Difference}$$

$$\text{grad AC} = \frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2 \cdot \Delta S} = \text{Central Difference}$$

$$\text{grad BD} = \frac{V(S + \Delta S, t) - V(S, t)}{\Delta S} = \text{Forward Difference}$$

Even from eyeballing the diagram it is easy to see that the central difference appears to be the most accurate. In the coming discussion we will see that this, is indeed, the case mathematically. We will also see that even though they are less accurate, in certain circumstances it is preferable to use either a forward or backward difference estimator.

### 3.1 Theta

Consider the 1<sup>st</sup> order Taylor approximation of  $V(S, t)$  around  $t$ :

$$V(S, t + \Delta t) = V(S, t) + \frac{1}{1!} \cdot \Delta t \cdot \frac{\partial V}{\partial t} + O[(\Delta t)^2]$$

where  $O[(\Delta t)^2]$  represents all terms of order  $(\Delta t)^2$  and higher. This can be simplified to:

$$V(S, t + \Delta t) - V(S, t) = \Delta t \cdot \frac{\partial V}{\partial t} + O[(\Delta t)^2]$$

which, after dividing both sides by  $\Delta t$  gives:

$$\frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} = \frac{\partial V}{\partial t} + O[\Delta t]$$

ie.<sup>4</sup>

$$\frac{\partial V}{\partial t} = \frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} + O[\Delta t]$$

or:

---

<sup>4</sup> Note that the sign of  $O[\Delta t]$  does not change as it is implicit within the function  $O[.]$ .

$$\frac{\partial V}{\partial t} \approx \frac{V(S, t + \Delta t) - V(S, t)}{\Delta t} \dots\dots\dots(3.1)$$

which is a **Forward Difference** estimate for **Theta**  $\left(\frac{\partial V}{\partial t}\right)$ .

A Backward Difference estimate is obtained by expanding with  $-\Delta t$  instead of  $+\Delta t$ , ie.:

$$V(S, t + (-\Delta t)) = V(S, t) + \frac{1}{1!} \cdot (-\Delta t) \cdot \frac{\partial V}{\partial t} + O[(-\Delta t)^2]$$

ie.

$$V(S, t - \Delta t) = V(S, t) - \frac{1}{1!} \cdot \Delta t \cdot \frac{\partial V}{\partial t} + O[(\Delta t)^2]$$

ie.

$$V(S, t - \Delta t) - V(S, t) = -\Delta t \cdot \frac{\partial V}{\partial t} + O[(\Delta t)^2]$$

which, when dividing both sides by  $-\Delta t$  becomes:

$$\frac{V(S, t - \Delta t) - V(S, t)}{-\Delta t} = \frac{\partial V}{\partial t} + O[\Delta t]$$

ie.

$$\frac{V(S, t) - V(S, t - \Delta t)}{\Delta t} = \frac{\partial V}{\partial t} + O[\Delta t]$$

ie.

$$\frac{\partial V}{\partial t} = \frac{V(S, t) - V(S, t - \Delta t)}{\Delta t} + O[\Delta t]$$

ie.

$$\frac{\partial V}{\partial t} \approx \frac{V(S, t) - V(S, t - \Delta t)}{\Delta t} \dots\dots\dots(3.2)$$

which is a **Backward Difference** estimate for **Theta**.

Note that both the Forward and Backward Difference estimators produce an error of the same order,  $O[\Delta t]$ .

### 3.2 Delta

Now consider the 3<sup>rd</sup> order Taylor approximation of  $V(S, t)$  around  $S$ :

$$V(S + \Delta S, t) = V(S, t) + \frac{1}{1!} \cdot \Delta S \cdot \frac{\partial V}{\partial S} + \frac{1}{2!} \cdot (\Delta S)^2 \cdot \frac{\partial^2 V}{\partial S^2} + \frac{1}{3!} \cdot (\Delta S)^3 \cdot \frac{\partial^3 V}{\partial S^3} + O[(\Delta S)^4]$$

ie.

$$V(S + \Delta S, t) = V(S, t) + \Delta S \cdot \frac{\partial V}{\partial S} + \frac{(\Delta S)^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} + \frac{(\Delta S)^3}{6} \cdot \frac{\partial^3 V}{\partial S^3} + O[(\Delta S)^4] \quad \dots\dots\dots(3.3)$$

Where  $O[(\Delta S)^4]$  represents all terms of order  $(\Delta S)^4$  and higher. Similarly, for  $-\Delta S$  we have:

$$V(S + (-\Delta S), t) = V(S, t) + (-\Delta S) \cdot \frac{\partial V}{\partial S} + \frac{(-\Delta S)^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} + \frac{(-\Delta S)^3}{6} \cdot \frac{\partial^3 V}{\partial S^3} + O[(-\Delta S)^4]$$

ie:

$$V(S - \Delta S, t) = V(S, t) - \Delta S \cdot \frac{\partial V}{\partial S} + \frac{(\Delta S)^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} - \frac{(\Delta S)^3}{6} \cdot \frac{\partial^3 V}{\partial S^3} + O[(\Delta S)^4] \quad \dots\dots\dots(3.4)$$

Subtracting equation 3.4 from equation 3.3 gives:

$$V(S + \Delta S, t) - V(S - \Delta S, t) = 2 \cdot \Delta S \cdot \frac{\partial V}{\partial S} + \left\{ \frac{(\Delta S)^3}{3} \cdot \frac{\partial^3 V}{\partial S^3} + O[(\Delta S)^4] \right\}$$

Simplifying the braced terms to  $O[(\Delta S)^3]$  gives:

$$V(S + \Delta S, t) - V(S - \Delta S, t) = 2 \cdot \Delta S \cdot \frac{\partial V}{\partial S} + O[(\Delta S)^3]$$

and dividing both sides by  $2 \cdot \Delta S$ :

$$\frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2 \cdot \Delta S} = \frac{\partial V}{\partial S} + O[(\Delta S)^2]$$

ie.

$$\frac{\partial V}{\partial S} = \frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2 \cdot \Delta S} + O[(\Delta S)^2]$$

or:

$$\frac{\partial V}{\partial S} \approx \frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2 \cdot \Delta S} \quad \dots\dots\dots(3.5)$$

which is a **Central Difference** estimator for **Delta**  $\left( \frac{\partial V}{\partial S} \right)$ .



Note that the central difference estimator has an error of the order  $O[(\Delta S)^2]$ . We could have chosen to use a central difference estimator for Theta rather than a backward or forward difference (see previously). This would result in a smaller error for our estimate of Theta as  $O[(\Delta t)^2] < O[\Delta t]$ . However, for reasons that will become apparent later, it is better to use forward or backward difference estimators for Theta.

### 3.3 Gamma

To find a Central Difference estimator for Gamma we add equation 3.4 to equation 3.3, rather than subtract it (as for Delta). This gives a different equation:

$$V(S + \Delta S, t) + V(S - \Delta S, t) = 2 \cdot V(S, t) + (\Delta S)^2 \cdot \frac{\partial^2 V}{\partial S^2} + O[(\Delta S)^3]$$

ie.

$$V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t) = (\Delta S)^2 \cdot \frac{\partial^2 V}{\partial S^2} + O[(\Delta S)^3]$$

ie.

$$\frac{V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)}{(\Delta S)^2} = \frac{\partial^2 V}{\partial S^2} + O[(\Delta S)^3]$$

ie.

$$\frac{\partial^2 V}{\partial S^2} = \frac{V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)}{(\Delta S)^2} + O[(\Delta S)^3]$$

or:

$$\frac{\partial^2 V}{\partial S^2} \approx \frac{V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)}{(\Delta S)^2} \dots\dots\dots(3.6)$$

which is a **Central Difference** estimator for **Gamma**  $\left( \frac{\partial^2 V}{\partial S^2} \right)$ .

### 4. The Explicit Euler Method

Recall the Black-Scholes PDE (equation 2.5):

$$\frac{\partial V}{\partial t} + r \cdot S \cdot \frac{\partial V}{\partial S} + \frac{S^2 \cdot \sigma^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} - r \cdot V = 0$$

If we use central difference estimators for Delta and Gamma and use a *backward difference* estimator for Theta, we can write the equation in terms of finite differences as follows:

$$\left[ \frac{V(S, t) - V(S, t - \Delta t)}{\Delta t} \right] + r \cdot S \cdot \left[ \frac{V(S + \Delta S, t) - V(S - \Delta S, t)}{2 \cdot \Delta S} \right] + \frac{S^2 \cdot \sigma^2}{2} \cdot \left[ \frac{V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)}{(\Delta S)^2} \right] - r \cdot V(S, t) = O[\Delta t, (\Delta S)^2]$$

Multiplying both sides by  $\Delta t$  and moving denominators to the coefficient terms gives:

$$[V(S, t) - V(S, t - \Delta t)] + \frac{r \cdot S \cdot \Delta t}{2 \cdot \Delta S} \cdot [V(S + \Delta S, t) - V(S - \Delta S, t)] + \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} \cdot [V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)] - r \cdot \Delta t \cdot V(S, t) = O[(\Delta t)^2, \Delta t \cdot (\Delta S)^2]$$

ie.:

$$\begin{aligned} V(S, t - \Delta t) = & V(S, t) \\ & + \frac{r \cdot S \cdot \Delta t}{2 \cdot \Delta S} \cdot [V(S + \Delta S, t) - V(S - \Delta S, t)] \\ & + \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} \cdot [V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)] \\ & - r \cdot \Delta t \cdot V(S, t) \\ & + O[(\Delta t)^2, \Delta t \cdot (\Delta S)^2] \end{aligned}$$

Grouping similar  $V(\dots)$  terms:

$$\begin{aligned} V(S, t - \Delta t) = & + \left[ \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} - \frac{r \cdot S \cdot \Delta t}{2 \cdot \Delta S} \right] \cdot V(S - \Delta S, t) \\ & + \left[ 1 - \frac{2 \cdot S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} - r \cdot \Delta t \right] \cdot V(S, t) \\ & + \left[ \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} + \frac{r \cdot S \cdot \Delta t}{2 \cdot \Delta S} \right] \cdot V(S + \Delta S, t) \\ & + O[(\Delta t)^2, \Delta t \cdot (\Delta S)^2] \end{aligned}$$

Dropping the  $O[\dots]$  operator and simplifying:

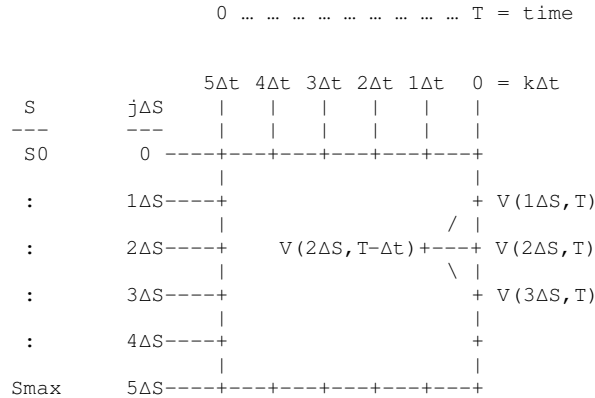
$$\begin{aligned} V(S, t - \Delta t) \approx & + \left[ \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} - \frac{r \cdot S \cdot \Delta t}{2 \Delta S} \right] \cdot V(S - \Delta S, t) \\ & + \left[ 1 - \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{(\Delta S)^2} - r \cdot \Delta t \right] \cdot V(S, t) \quad \dots\dots\dots(4.1) \\ & + \left[ \frac{S^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} + \frac{r \cdot S \cdot \Delta t}{2 \cdot \Delta S} \right] \cdot V(S + \Delta S, t) \end{aligned}$$

Which is an expression of  $V(\dots, t - \Delta t)$  in terms of  $V(\dots, t)$ .

---

## 4.1 An Algorithm for Explicit Euler

To recap: we have taken the Black-Scholes PDE, expressed it in terms of finite differences and rearranged it to express an option's value at time  $t - \Delta t$  as a function of its value at time  $t$ . We express this graphically using some ASCII-art. Consider the following example:

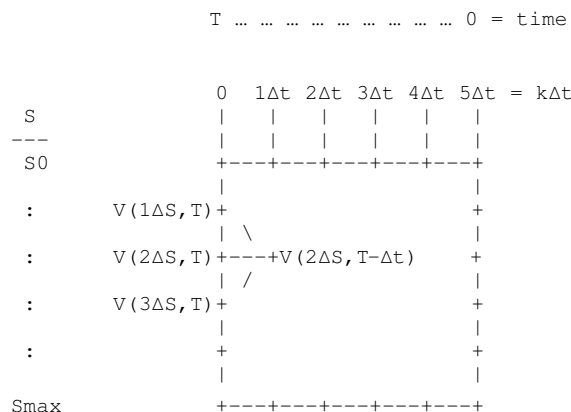


While this diagram may seem daunting at first, upon further analysis it's quite simple. Down the left hand side we have discretized the asset price  $S$  into  $j$  units of  $\Delta S$  each ( $j=5$  in this example) ranging from  $S=0$  up to  $S=S_{\max}$ . Starting from the top left we see that  $S$  starts at zero and increases as we move down the axis to  $5\Delta S$ .

The time axis, however, is not so straight forward. It starts from the right ( $time=T$ ) and works backwards in time to the left with each node ('-+-') representing a movement back in time by  $\Delta t$  until we reach  $time=0$  ( $k=5$  in this example). This representation of time is in keeping with our choice of a *backwards* finite difference estimator for Theta.

Shown in the diagram is our finite difference estimate  $V(2\Delta S, T-\Delta t)$  which is in accordance with equation 3.1 (coefficients are not shown, of course). Therefore, if we repeat this calculation for all other nodes at  $time = T-\Delta t$  then we build a grid – column by column – as we move backwards to  $time=0$ . This, in essence, is the Explicit Euler Method of finite difference estimation: iteratively working our way back from a final position to an initial position.

While such an approach is easily understood, it is often the case that in textbooks that our final condition is turned into an initial condition by reversing the time variable so that we start from the left and work our way, iteratively, across to the right. We are still, of course, moving back in time but we are doing so from left to right, rather than right to left. The following diagram shows this:



Now, we have both  $S$  and  $t$  starting at the top left. So, moving from left to right on the time axis, we start at the option's expiry date  $T$  and move forward (back in time) to the option's start date. It is  $V$  at this date ie.  $V(\cdot, 0)$  that is the whole purpose of the method of finite differences: we are trying to estimate the option's price at  $t_{\text{time}}=0$ .

Taking equation 4.1 and expressing its  $S$  and  $t$  variables in our discretized form gives:

$$\begin{aligned}
 V(j \cdot \Delta S, [k+1], \Delta t) \approx & \\
 & + \left[ \frac{(j \cdot \Delta S)^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} - \frac{r \cdot j \cdot \Delta S \cdot \Delta t}{2 \cdot \Delta S} \right] \cdot V([j-1] \cdot \Delta S, k, \Delta t) \\
 & + \left[ 1 - \frac{(j \cdot \Delta S)^2 \cdot \sigma^2 \cdot \Delta t}{(\Delta S)^2} - r \cdot \Delta t \right] \cdot V(j \cdot \Delta S, k, \Delta t) \\
 & + \left[ \frac{(j \cdot \Delta S)^2 \cdot \sigma^2 \cdot \Delta t}{2 \cdot (\Delta S)^2} + \frac{r \cdot j \cdot \Delta S \cdot \Delta t}{2 \cdot \Delta S} \right] \cdot V([j+1] \cdot \Delta S, k, \Delta t)
 \end{aligned}$$

Simplifying the coefficients gives:

$$\begin{aligned}
 V(j \cdot \Delta S, [k+1], \Delta t) \approx & \\
 & + \left[ \frac{j^2 \cdot \sigma^2 \cdot \Delta t}{2} - \frac{r \cdot j \cdot \Delta t}{2} \right] \cdot V([j-1] \cdot \Delta S, k, \Delta t) \\
 & + \left[ 1 - \frac{j^2 \cdot \sigma^2 \cdot \Delta t}{1} - r \cdot \Delta t \right] \cdot V(j \cdot \Delta S, k, \Delta t) \\
 & + \left[ \frac{j^2 \cdot \sigma^2 \cdot \Delta t}{2} + \frac{r \cdot j \cdot \Delta t}{2} \right] \cdot V([j+1] \cdot \Delta S, k, \Delta t)
 \end{aligned}$$

and bringing out common factors gives:

$$\begin{aligned}
 V(j \Delta S, [k+1], \Delta t) \approx & \\
 & + \frac{j \Delta t}{2} \cdot [j \cdot \sigma^2 - r] \cdot V([j-1] \cdot \Delta S, k, \Delta t) \\
 & + [1 - j^2 \cdot \sigma^2 \cdot \Delta t - r \cdot \Delta t] \cdot V(j \cdot \Delta S, k, \Delta t) \\
 & + \frac{j \Delta t}{2} \cdot [j \cdot \sigma^2 + r] \cdot V([j+1] \cdot \Delta S, k, \Delta t)
 \end{aligned}$$

ie.

$$\begin{aligned}
 V(j \Delta S, [k+1], \Delta t) \approx & \\
 & + a_j \cdot V([j-1] \cdot \Delta S, k, \Delta t) \\
 & + b_j \cdot V(j \cdot \Delta S, k, \Delta t) \\
 & + c_j \cdot V([j+1] \cdot \Delta S, k, \Delta t)
 \end{aligned}$$

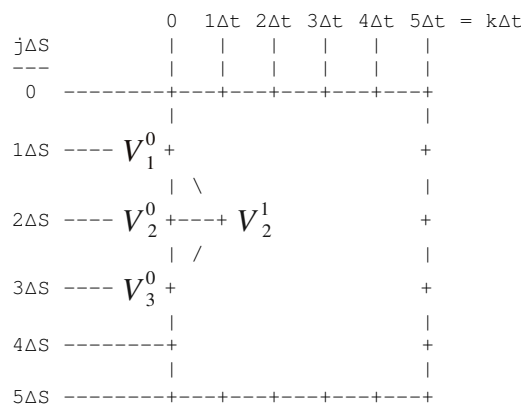
where<sup>5</sup>

$$\begin{aligned}
 a_j &= \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 - r] \\
 b_j &= [1 - j^2 \cdot \sigma^2 \cdot \Delta t - r \cdot \Delta t] \dots\dots\dots(4.2) \\
 c_j &= \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 + r]
 \end{aligned}$$

This can be expressed in simpler notation by using  $V_j^k = V(j, \Delta S, k, \Delta t)$  such that the above equation becomes:

$$V_j^{k+1} \approx a_j \cdot V_{j-1}^k + b_j \cdot V_j^k + c_j \cdot V_{j+1}^k \dots\dots\dots(4.3)$$

Which is the form often seen in textbooks and can be visualised like so:



Recall equation 4.3:

$$V_j^{k+1} \approx a_j \cdot V_{j-1}^k + b_j \cdot V_j^k + c_j \cdot V_{j+1}^k$$

Using our previous example (j=0, 1, ..., 5 and k=0, 1, ..., 5) this gives general equation rise to the following system of linear equations:

$$\begin{aligned}
 V_1^1 &\approx a_1 \cdot V_0^0 + b_1 \cdot V_1^0 + c_1 \cdot V_2^0 \\
 V_2^1 &\approx a_2 \cdot V_1^0 + b_2 \cdot V_2^0 + c_2 \cdot V_3^0 \\
 V_3^1 &\approx a_3 \cdot V_2^0 + b_3 \cdot V_3^0 + c_3 \cdot V_4^0 \\
 V_4^1 &\approx a_4 \cdot V_3^0 + b_4 \cdot V_4^0 + c_4 \cdot V_5^0
 \end{aligned}$$

which, when flipped around the equalities, can be represented in matrix form  $Ax=b$  as:

---

<sup>5</sup> Note that none of these coefficients is a function of k. This has important consequences for our algorithm which we will discover shortly.

$$\begin{bmatrix} a_1 & b_1 & c_1 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & 0 \\ 0 & 0 & 0 & a_4 & b_4 & c_4 \end{bmatrix} \cdot \begin{bmatrix} V_0^0 \\ V_1^0 \\ V_2^0 \\ V_3^0 \\ V_4^0 \\ V_5^0 \end{bmatrix} = \begin{bmatrix} V_1^1 \\ V_2^1 \\ V_3^1 \\ V_4^1 \end{bmatrix}$$

However, on the next iteration (the next time step) a problem becomes immediately apparent:

$$\begin{bmatrix} a_1 & b_1 & c_1 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & 0 \\ 0 & 0 & 0 & a_4 & b_4 & c_4 \end{bmatrix} \cdot \begin{bmatrix} V_0^1 \\ V_1^1 \\ V_2^1 \\ V_3^1 \\ V_4^1 \\ V_5^1 \end{bmatrix} = \begin{bmatrix} V_1^2 \\ V_2^2 \\ V_3^2 \\ V_4^2 \end{bmatrix}$$

We cannot determine either  $V_1^2$  or  $V_4^2$  because we do not know  $V_0^1$  and  $V_5^1$ . Looking at the previous iteration we see that  $V_0^1$  and  $V_5^1$  were never determined. In short, we have not specified the conditions at the boundaries of the region where our partial differential equation is defined.

What these *boundary conditions* are will depend upon the nature of the option contract. For a *European Call* option, they are usually chosen based on the the following observations:

$$\text{as } S \rightarrow 0 \text{ then } V_{Call} \rightarrow 0 \text{ AND as } S \rightarrow \infty \text{ then } V_{Call} \rightarrow S$$

The former states that as  $S$  gets closer and closer to zero – ie. is deeper and deeper out-of-the-money – the probability that it will go above the call option's strike price (during the remaining term of the option) gets smaller and smaller. There comes a point where this probability is so low as to be negligible. Hence, the value of the call option is zero.

The latter observation specifies the opposite: as  $S$  gets larger and larger relative to the strike price the option begins to behave more and more like the underlying. For very large  $S$ , the option behaves just like  $S$ . However, in order to avoid excessive computation time caused by a large number of asset steps, it is desirable to use a different boundary condition:

---

$$\text{as } S \rightarrow j \cdot \Delta S \text{ then } V_{Call} \rightarrow j \cdot \Delta S - K \cdot e^{-r \cdot k \cdot \Delta t}$$

where  $K$  is the option's exercise price.

For a European Put option the boundary conditions are usually set based on:

$$S \rightarrow 0 \text{ as then } V_{Put} \rightarrow K \cdot e^{-r \cdot (T-t)} \text{ AND } S \rightarrow \infty \text{ as then } V_{Put} \rightarrow 0$$

These, however, are not the only boundary conditions that may be imposed. An alternative is to look at the PDE itself and see how it behaves when  $S \rightarrow 0$  and  $S \rightarrow \infty$ . Recall the Black-Scholes PDE (equation 2.5):

$$\frac{\partial V}{\partial t} + r \cdot S \cdot \frac{\partial V}{\partial S} + \frac{S^2 \cdot \sigma^2}{2} \cdot \frac{\partial^2 V}{\partial S^2} - r \cdot V = 0$$

As  $S \rightarrow 0$  we can see that the equation becomes:

$$\frac{\partial V}{\partial t} - r \cdot V = 0$$

which, when substituting our backwards difference estimator for Theta (equation 3.2) becomes:

$$\frac{V(0, t) - V(0, t - \Delta t)}{\Delta t} - r \cdot V(0, t) \approx 0$$

ie.:

$$V(0, t) - r \cdot \Delta t \cdot V(0, t) \approx V(0, t - \Delta t)$$

ie.:

$$[1 - r \cdot \Delta t] \cdot V(0, t) \approx V(0, t - \Delta t)$$

which, using our discretized notation is:

$$[1 - r \cdot \Delta t] \cdot V_0^k \approx V_0^{k+1}$$

ie.:

$$V_0^{k+1} \approx [1 - r \cdot \Delta t] \cdot V_0^k$$

ie.:

$$V_0^k \approx [1 - r \cdot \Delta t] \cdot V_0^{k-1} \dots \dots \dots (4.4)$$

As  $S \rightarrow \infty$  then  $\frac{\partial^2 V}{\partial S^2} \rightarrow 0$ , ie. as  $S$  gets very large the option value becomes linear such that Gamma

becomes zero. Using our central difference estimator for Gamma (equation 3.5) gives:

$$\frac{V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t)}{(\Delta S)^2} \approx 0$$

ie.

$$V(S + \Delta S, t) - 2 \cdot V(S, t) + V(S - \Delta S, t) \approx 0$$

ie.

$$V(S + \Delta S, t) \approx 2 \cdot V(S, t) - V(S - \Delta S, t)$$

which, when discretized becomes:

$$V_{j+1}^k \approx 2 \cdot V_j^k - V_{j-1}^k$$

ie.:

$$V_j^k \approx 2 \cdot V_{j-1}^k - V_{j-2}^k \dots \dots \dots (4.5)$$

Substituting equations for  $V_{j-1}^k$  and  $V_{j-2}^k$  gives:

$$V_j^k \approx 2 \cdot [a_{j-1} \cdot V_{j-2}^{k-1} + b_{j-1} \cdot V_{j-1}^{k-1} + c_{j-1} \cdot V_j^{k-1}] - [a_{j-2} \cdot V_{j-3}^{k-1} + b_{j-2} \cdot V_{j-2}^{k-1} + c_{j-2} \cdot V_{j-1}^{k-1}]$$

ie.:

$$V_j^k \approx -a_{j-2} \cdot V_{j-3}^{k-1} + (2 \cdot a_{j-1} - b_{j-2}) \cdot V_{j-2}^{k-1} + (2 \cdot b_{j-1} - c_{j-2}) \cdot V_{j-1}^{k-1} + c_{j-1} \cdot V_j^{k-1}$$

The beauty of these two alternative boundary conditions (ie. equations 4.4 and 4.5) is that they can be used for either a European Call or a European Put, thereby simplifying the coding.

Using our boundary conditions our linear system becomes:

$$\begin{bmatrix} (1-r \cdot \Delta t) & 0 & 0 & 0 & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & 0 & 0 \\ 0 & a_2 & b_2 & c_2 & 0 & 0 \\ 0 & 0 & a_3 & b_3 & c_3 & 0 \\ 0 & 0 & 0 & a_4 & b_4 & c_4 \\ 0 & 0 & (-a_3) & (2 \cdot a_4 - b_3) & (2 \cdot b_4 - c_3) & c_4 \end{bmatrix} \cdot \begin{bmatrix} V_0^0 \\ V_1^0 \\ V_2^0 \\ V_3^0 \\ V_4^0 \\ V_5^0 \end{bmatrix} = \begin{bmatrix} V_0^1 \\ V_1^1 \\ V_2^1 \\ V_3^1 \\ V_4^1 \\ V_5^1 \end{bmatrix}$$

which is now fully specified and can be coded as follows:

---



- i) set up the matrix  $A$ ;
- ii) use the option's payoff function (at option expiry:  $\text{time}=T$ ;  $k=0$ ) to establish vector  $\mathbf{x}$ ;
- iii) solve for vector  $\mathbf{b}$ ;
- iv) let  $\mathbf{x} = \mathbf{b}$ ;
- v) repeat iii) to iv) for the number of time steps<sup>6</sup>.

## 4.2 The Stability of Explicit Euler

The reader will have noted from the preceding discussion that finite difference methods give a series of estimates for  $V$  at each point in time: one for each value of  $S$ . So, by choosing the size of the asset step,  $\Delta S$ , the user can obtain the option value for a desired price level. This leads us to the main problem with Explicit Euler: stability.

Despite its simplicity, Explicit Euler is a BAD method. This is because its estimates become unstable when the number of time steps specified ( $k$ ) is less than the square of the number of asset steps specified ( $j$ ). Alternatively: when the size of the time step,  $\Delta t$ , exceeds a given value of the asset step,  $\Delta S$ . The result is that the Explicit Euler estimates “explode” (rather than converge) as we step through time.

Proof is by way of a concept in physics known as von Neumann stability. A finite difference method generating estimates  $V_j^k$  is von Neumann stable if, under the substitution  $V_j^k = \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t}$ , the absolute value of  $\xi$  is less than one, ie. if  $|\xi| < 1$ , where  $i = \sqrt{-1}$  and  $\beta \in [-\pi, \pi]$ .

So, substituting into equation 4.3 gives:

$$\xi^{k+1} \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} = a_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot (j-1) \cdot \Delta t} + b_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} + c_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot (j+1) \cdot \Delta t}$$

ie.

$$\xi^k \cdot \xi^1 \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} = a_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} \cdot e^{i \cdot \beta \cdot -1 \cdot \Delta t} + b_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} + c_j \cdot \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} \cdot e^{i \cdot \beta \cdot 1 \cdot \Delta t}$$

ie.

$$\xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} \cdot [\xi^1] = \xi^k \cdot e^{i \cdot \beta \cdot j \cdot \Delta t} \cdot [a_j \cdot e^{-i \cdot \beta \cdot \Delta t} + b_j + c_j \cdot e^{i \cdot \beta \cdot \Delta t}]$$

ie.

$$\xi = a_j \cdot e^{-i \cdot \beta \cdot \Delta t} + b_j + c_j \cdot e^{i \cdot \beta \cdot \Delta t}$$

Recognising that  $e^{-i \cdot x} = \cos(x) - i \cdot \sin(x)$  and  $e^{i \cdot x} = \cos(x) + i \cdot \sin(x)$  gives:

$$\xi = a_j \cdot [\cos(\beta \cdot \Delta t) - i \cdot \sin(\beta \cdot \Delta t)] + b_j + c_j \cdot [\cos(\beta \cdot \Delta t) + i \cdot \sin(\beta \cdot \Delta t)]$$

which rearranges to:

---

<sup>6</sup> The astute reader will notice that under these boundary conditions, we have a Markov Chain with  $A$  as the transition matrix. Consequently,  $A$  only needs to be established once leading to flexible and simplified coding.

$$\xi = b_j + [c_j + a_j] \cdot \cos(\beta \cdot \Delta t) + [c_j - a_j] \cdot i \cdot \sin(\beta \cdot \Delta t)$$

substituting in our values for  $a_j$ ,  $b_j$  and  $c_j$  (from equation 4.2) gives:

$$\begin{aligned} \xi = & \\ & + [1 - j^2 \cdot \sigma^2 \cdot \Delta t - r \cdot \Delta t] \\ & + \left[ \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 + r] + \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 - r] \right] \cdot \cos(\beta \cdot \Delta t) \\ & + \left[ \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 + r] - \frac{j \cdot \Delta t}{2} \cdot [j \cdot \sigma^2 - r] \right] \cdot i \cdot \sin(\beta \cdot \Delta t) \end{aligned}$$

which simplifies to:

$$\begin{aligned} \xi = & \\ & + [1 - j^2 \cdot \sigma^2 \cdot \Delta t - r \cdot \Delta t] \\ & + [j^2 \cdot \sigma^2 \cdot \Delta t] \cdot \cos(\beta \cdot \Delta t) \\ & + [j \cdot r \cdot \Delta t] \cdot \sqrt{-\sin^2(\beta \cdot \Delta t)} \end{aligned}$$

It turns out that for  $|\xi| < 1$  the following three conditions must hold:

$$r \cdot \Delta t \geq 0 \dots\dots\dots(4.6)$$

$$j^2 \cdot \sigma^2 \cdot \Delta t \geq j \cdot r \cdot \Delta t \dots\dots\dots(4.7)$$

$$1 - j^2 \cdot \sigma^2 \cdot \Delta t - r \cdot \Delta t \geq 0 \dots\dots\dots(4.8)$$

While equation 4.6 is clearly satisfied, equation 4.7 can be simplified to:

$$j^2 \cdot \sigma^2 \cdot \Delta t \geq j \cdot r \cdot \Delta t$$

$$j \cdot \sigma^2 \geq r$$

$$j \geq \frac{r}{\sigma^2}$$

which places a lower bound on the number of asset steps,  $j$ .

Equation 4.8 can be simplified to:

$$1 \geq j^2 \cdot \sigma^2 \cdot \Delta t + r \cdot \Delta t$$

$$1 \geq [j^2 \cdot \sigma^2 + r] \cdot \Delta t$$

$$\frac{1}{\Delta t} \geq j^2 \cdot \sigma^2 + r$$

and since  $\Delta t = \frac{T}{k}$ :

$$\frac{k}{T} \geq j^2 \cdot \sigma^2 + r \dots\dots\dots(4.9)$$

This equation is the one which best demonstrates why Explicit Euler is viewed by some as a bad method. If we increase the number of asset steps – in order to increase our precision on  $S$  – then we must increase the number of time steps by the *square* of the asset step increase. For example if we increase  $j$  by a factor of 2 (ie. we double the number of asset steps) then equation 4.9 requires us to increase  $k$  by a factor of  $2^2$  (ie. we quadruple the number of time steps), so our computation time goes up by a factor of 8!

### 4.3 Example VBA Code

Having developed our algorithm and identified our stability constraints it is time to write some code. The following code listing is a simple but thread-bare Explicit Euler model that incorporates what has been discussed so far.

```

1      Public Function ExplicitEuler( _
2          sglStrike As Single, _
3          sglSigma As Single, _
4          sglRfrate As Single, _
5          sglTerm As Single, _
6          intNAS As Integer, _
7          Optional strType As String = "Call" _
8      ) As Variant
9
10     Dim j As Integer, _
11         k As Integer, _
12         intNTS As Integer, _
13         sglDeltaS As Single, _
14         sglDeltaT As Single, _
15         sglarrTM() As Single, _
16         vararrVold As Variant, _
17         vararrVnew As Variant
18
19     '// ensure stability of Explicit Euler
20     intNAS = Application.WorksheetFunction.Max(intNAS, Int(sglRfrate / sglSigma ^ 2) + 1) '// satisfy constraint on # asset steps
21     intNTS = Int((intNAS ^ 2 * sglSigma ^ 2 + sglRfrate) * sglTerm) + 1 '// satisfy constraint on # time steps
22
23     '// determine size of steps
24     sglDeltaS = 2 * sglStrike / intNAS '// note: Smax = 2 x Strike Price
25     sglDeltaT = sglTerm / intNTS
26
27     '// redimension Transition Matrix (TM) and vectors of option value estimates (V-old an V-new)
28     ReDim sglarrTM(0 To intNAS, 0 To intNAS), _
29         vararrVold(0 To intNAS, 0 To 0), _
30         vararrVnew(0 To intNAS, 0 To 0)
31
32     '// place FD coefficients (aj, bj and cj) into Transition Matrix
33     For j = 1 To intNAS - 1
34         sglarrTM(j, j - 1) = j * sglDeltaT / 2 * (j * sglSigma ^ 2 - sglRfrate) '// aj
35         sglarrTM(j, j) = 1 - j ^ 2 * sglSigma ^ 2 * sglDeltaT - sglRfrate * sglDeltaT '// bj
36         sglarrTM(j, j + 1) = j * sglDeltaT / 2 * (j * sglSigma ^ 2 + sglRfrate) '// cj
37     Next j
38
39     '// set boundary conditions in Transition Matrix:
40     '// as S->0
41     sglarrTM(0, 0) = 1 - sglRfrate * sglDeltaT
42     '// as S -> Smax
43     sglarrTM(intNAS, intNAS - 3) = -1 * sglarrTM(intNAS - 2, intNAS - 3)
44     sglarrTM(intNAS, intNAS - 2) = 2 * sglarrTM(intNAS - 1, intNAS - 2) - sglarrTM(intNAS - 2, intNAS - 2)
45     sglarrTM(intNAS, intNAS - 1) = 2 * sglarrTM(intNAS - 1, intNAS - 1) - sglarrTM(intNAS - 2, intNAS - 1)
46     sglarrTM(intNAS, intNAS) = 2 * sglarrTM(intNAS - 1, intNAS)
47

```

```

48 // set initial condition (= option payoff at expiry) in column vector of option value estimates
49 For j = 0 To intNAS
50     vararrVold(j, 0) = Application.WorksheetFunction.Max(0, (j * sglDeltaS - sglStrike) * If(strType = "Put", -1, 1))
51 Next j
52
53 // now step through each time step (ie. going 'back in time') to reach our answer
54 For k = 1 To intNTS
55     vararrVnew = Application.WorksheetFunction.MMult(sglarrTM, vararrVold)
56     vararrVold = vararrVnew
57 Next k
58
59 ExplicitEuler = vararrVold
60
61 End Function

```

The following points can be made about the above code:

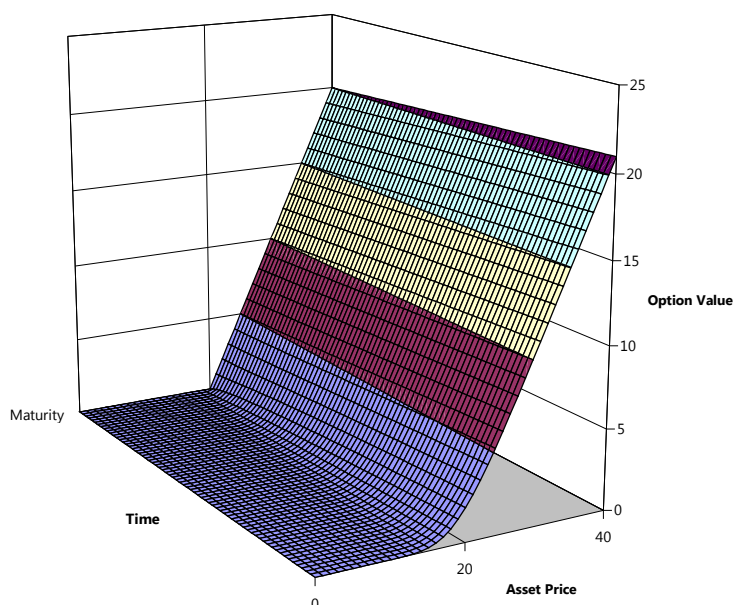
- Lines 7 and 50 provide flexibility to price either a put or call option.
- Lines 19 to 21 set the number of asset steps and the number of time steps so as to ensure stability of our estimator.
- Line 24 sets our proxy for  $S_{max}$  equal to double the option's strike price.
- Lines 39-46 implement our boundary conditions.
- Lines 48-51 gets the algorithm going by setting our initial condition.
- Lines 53-57 is the 'heavy lifting' ie. where we step back through time to find our initial option values.
- Line 59 returns our answer: a column vector of option value estimates at time=0.

#### 4.4 Example Numerical Results

The chart and tables below have been produced based on the following assumptions:

Strike Price	Volatility	Risk-free Rate	Term	# Asset Steps	Type
20	0.2	0.05	1	40	Call

A modified form of the code in section 4.3 was used which is listed in full at Appendix A.



In this chart we can clearly see how we started at the back (Maturity) and worked our way forward to time=0 to go from our rigid payoff-at-expiry function to the familiar two-dimensional 'hockey stick' curve of Black-Scholes as often seen in text books.

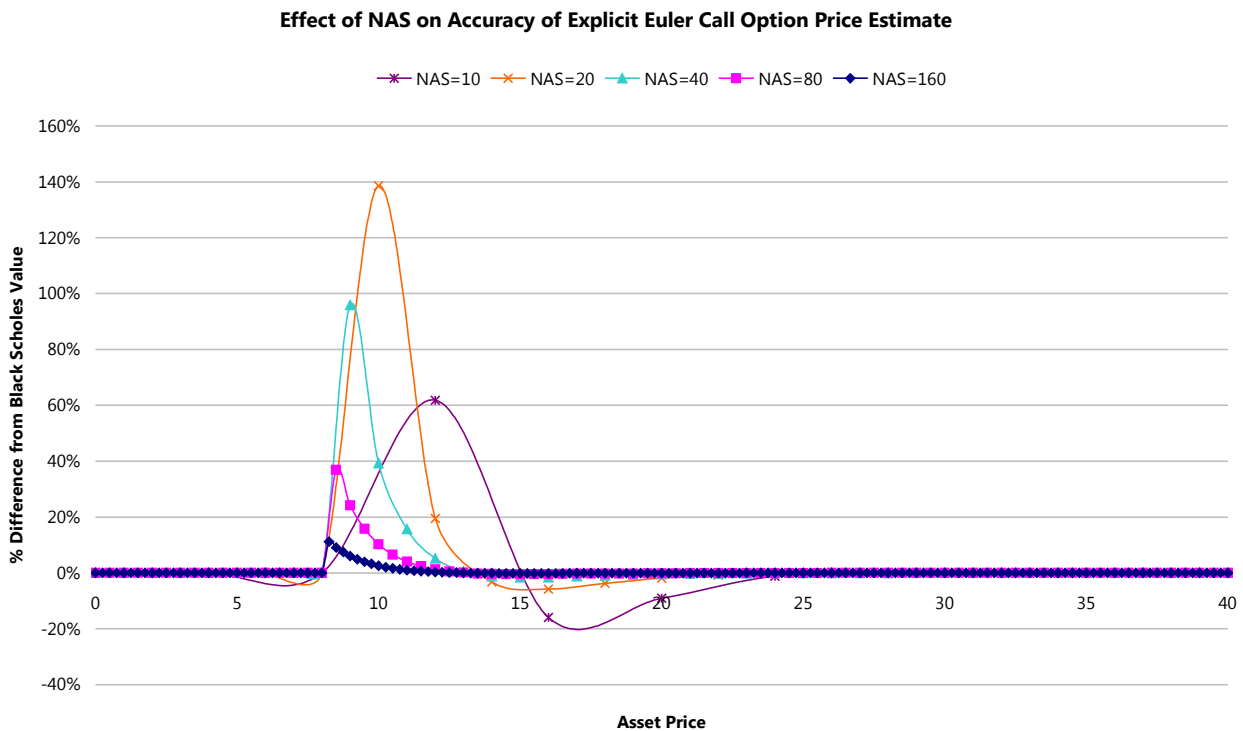
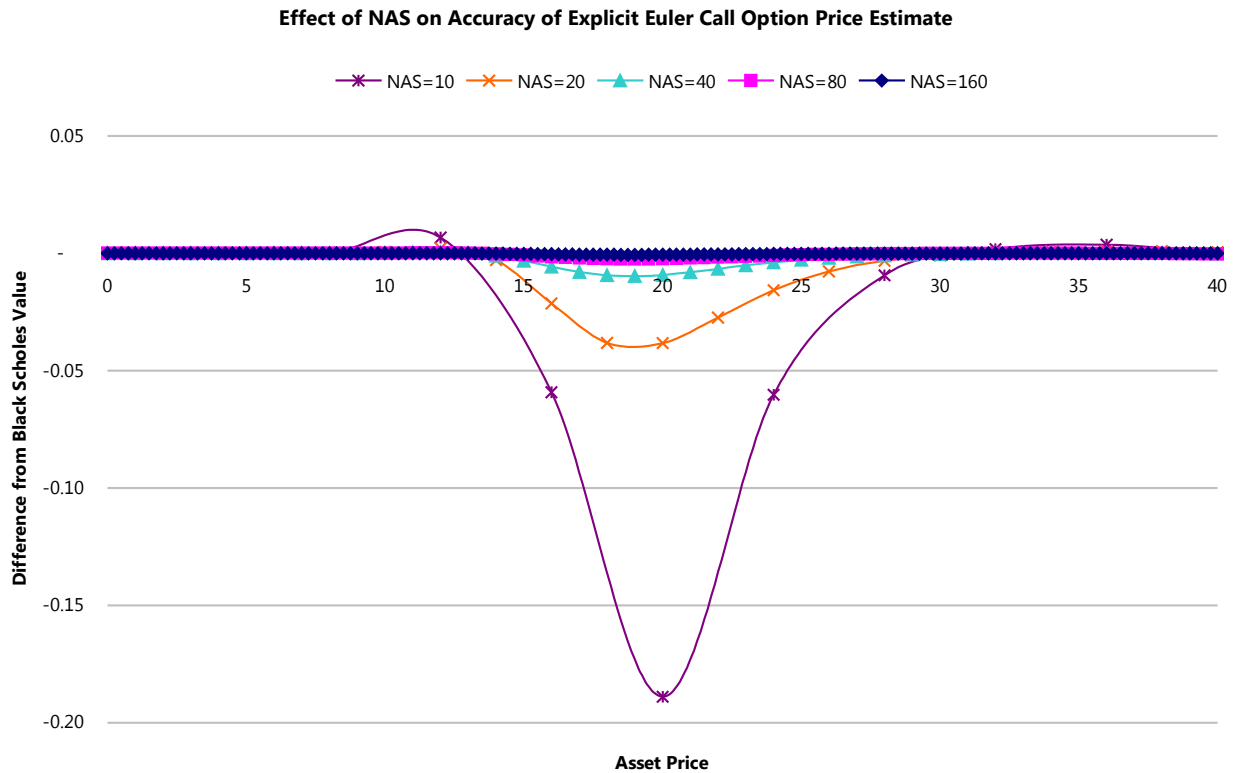
#### 4.4.1 Option Prices

In terms of its accuracy, the following table shows that Explicit Euler comes very close to the true Black-Scholes option price:

S	Explicit Euler Estimate	Black-Scholes Price	Difference	% Difference
0	0.00000	0.00000	0.00000	N/A
1	0.00000	0.00000	0.00000	N/A
2	0.00000	0.00000	0.00000	N/A
3	0.00000	0.00000	0.00000	N/A
4	0.00000	0.00000	0.00000	N/A
5	0.00000	0.00000	0.00000	N/A
6	0.00000	0.00000	0.00000	N/A
7	0.00000	0.00000	0.00000	N/A
8	0.00001	0.00000	0.00001	256.00%
9	0.00011	0.00006	0.00005	96.00%
10	0.00067	0.00048	0.00019	39.00%
11	0.00311	0.00269	0.00042	16.00%
12	0.01145	0.01089	0.00056	5.00%
13	0.03449	0.03428	0.00021	1.00%
14	0.08726	0.08829	-0.00103	-1.00%
15	0.19030	0.19345	-0.00314	-2.00%
16	0.36622	0.37188	-0.00566	-2.00%
17	0.63482	0.64272	-0.00790	-1.00%
18	1.00892	1.01824	-0.00932	-1.00%
19	1.49246	1.50217	-0.00971	-1.00%
20	2.08093	2.09012	-0.00919	0.00%
21	2.76353	2.77158	-0.00805	0.00%
22	3.52597	3.53259	-0.00663	0.00%
23	4.35292	4.35810	-0.00518	0.00%
24	5.22993	5.23381	-0.00388	0.00%
25	6.14441	6.14721	-0.00280	0.00%
26	7.08610	7.08805	-0.00195	0.00%
27	8.04704	8.04835	-0.00131	0.00%
28	9.02128	9.02212	-0.00084	0.00%
29	10.00453	10.00503	-0.00050	0.00%
30	10.99377	10.99403	-0.00026	0.00%
31	11.98694	11.98702	-0.00008	0.00%
32	12.98265	12.98260	0.00004	0.00%
33	13.97997	13.97984	0.00013	0.00%
34	14.97831	14.97813	0.00018	0.00%
35	15.97728	15.97707	0.00021	0.00%
36	16.97664	16.97642	0.00022	0.00%
37	17.97622	17.97602	0.00020	0.00%
38	18.97593	18.97578	0.00016	0.00%
39	19.97569	19.97563	0.00006	0.00%
40	20.97546	20.97554	-0.00009	0.00%

As can be seen, the differences are small in absolute terms but start to become quite large in relative terms (ie. % Difference) when the option is deeply out-of-the-money. These relative differences become so large that they were truncated for  $S < 8$ .

The precision of Explicit Euler can be improved by increasing the number of asset steps but, as we have seen, this comes at the cost of increased computation time. The following charts shows how accuracy improves as the number of asset steps (“NAS”) increases:



What is notable from the first chart is that while increasing NAS does increase accuracy, most inaccuracy occurs when  $S$  is near the strike price. Also striking is the consistent underpricing of

Explicit Euler over this region. The second chart shows what we have seen from our tabular results: that the large percentage values when  $S < 15$  are explained by the Black Scholes Values being close to zero when an option is deeply out-of-the-money.

The charts suggest NAS should be 40 or more in order to get reasonable accuracy for pricing purposes. Of course, depending on the notional value of the option contract under consideration, greater accuracy may be required.

#### 4.4.2 Delta and Gamma

The following table shows the accuracy of the Explicit Euler Delta and Gamma estimates (refer equations 3.5 and 3.6)<sup>7</sup>. Note that values have been truncated when  $S$  is deeply out-of-the-money and no estimates are produced for the end points due to the use of central differences<sup>8</sup>:

S	Delta				Gamma			
	Explicit Euler Estimate	Black Scholes Value	Difference	% Difference	Explicit Euler Estimate	Black Scholes Value	Difference	% Difference
0	N/A	0.00000	N/A	N/A	N/A	0.00000	N/A	N/A
1	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
2	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
3	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
4	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
5	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
6	0.00000	0.00000	0.00000	N/A	0.00000	0.00000	0.00000	N/A
7	0.00001	0.00000	0.00001	N/A	0.00001	0.00000	0.00001	547.00%
8	0.00005	0.00001	0.00004	372.00%	0.00008	0.00003	0.00005	162.00%
9	0.00033	0.00013	0.00019	143.00%	0.00046	0.00029	0.00017	58.00%
10	0.00150	0.00092	0.00058	63.00%	0.00188	0.00156	0.00032	21.00%
11	0.00539	0.00416	0.00124	30.00%	0.00591	0.00557	0.00034	6.00%
12	0.01569	0.01376	0.00193	14.00%	0.01469	0.01465	0.00005	0.00%
13	0.03790	0.03562	0.00228	6.00%	0.02973	0.03015	-0.00042	-1.00%
14	0.07791	0.07588	0.00203	3.00%	0.05028	0.05100	-0.00073	-1.00%
15	0.13948	0.13821	0.00128	1.00%	0.07288	0.07354	-0.00067	-1.00%
16	0.22226	0.22192	0.00034	0.00%	0.09267	0.09299	-0.00032	0.00%
17	0.32135	0.32183	-0.00048	0.00%	0.10551	0.10543	0.00008	0.00%
18	0.42882	0.42983	-0.00101	0.00%	0.10943	0.10910	0.00034	0.00%
19	0.53600	0.53726	-0.00126	0.00%	0.10493	0.10453	0.00040	0.00%
20	0.63554	0.63683	-0.00130	0.00%	0.09414	0.09381	0.00033	0.00%
21	0.72252	0.72373	-0.00121	0.00%	0.07983	0.07963	0.00020	0.00%
22	0.79470	0.79575	-0.00106	0.00%	0.06452	0.06443	0.00009	0.00%
23	0.85198	0.85287	-0.00089	0.00%	0.05005	0.05004	0.00001	0.00%
24	0.89574	0.89646	-0.00071	0.00%	0.03747	0.03750	-0.00003	0.00%
25	0.92809	0.92864	-0.00055	0.00%	0.02721	0.02725	-0.00004	0.00%
26	0.95132	0.95173	-0.00041	0.00%	0.01925	0.01929	-0.00004	0.00%
27	0.96759	0.96788	-0.00029	0.00%	0.01330	0.01333	-0.00003	0.00%
28	0.97875	0.97894	-0.00020	0.00%	0.00901	0.00903	-0.00002	0.00%
29	0.98625	0.98637	-0.00013	0.00%	0.00599	0.00601	-0.00002	0.00%
30	0.99121	0.99128	-0.00007	0.00%	0.00393	0.00394	-0.00001	0.00%
31	0.99444	0.99448	-0.00004	0.00%	0.00254	0.00255	-0.00001	0.00%
32	0.99651	0.99653	-0.00002	0.00%	0.00162	0.00163	-0.00001	-1.00%
33	0.99783	0.99784	-0.00001	0.00%	0.00102	0.00103	-0.00001	-1.00%
34	0.99866	0.99866	-0.00001	0.00%	0.00063	0.00065	-0.00001	-2.00%

<sup>7</sup> Black Scholes Values for Delta and Gamma were determined using the analytical formulae derived in Appendix B.

<sup>8</sup> Forward and backward differences could be used to produce estimates, of course, but it is unlikely they would be needed in practice.

S	Delta				Gamma			
	Explicit Euler Estimate	Black Scholes Value	Difference	% Difference	Explicit Euler Estimate	Black Scholes Value	Difference	% Difference
35	0.99917	0.99918	-0.00001	0.00%	0.00039	0.00040	-0.00002	-4.00%
36	0.99947	0.99950	-0.00003	0.00%	0.00023	0.00025	-0.00002	-9.00%
37	0.99965	0.99969	-0.00005	0.00%	0.00012	0.00015	-0.00003	-19.00%
38	0.99974	0.99981	-0.00008	0.00%	0.00005	0.00009	-0.00004	-45.00%
39	0.99976	0.99989	-0.00013	0.00%	0.00000	0.00006	-0.00006	-100.00%
40	N/A	0.99993	N/A	N/A	N/A	0.00003	N/A	N/A

We can see that Explicit Euler provides fairly accurate estimates of the true, analytical Black-Scholes values and they are certainly within boundaries for practical application on the trading floor.

## 5. Conclusion

Despite its inherent shortcomings (stability constraints which lead to increased computational time) Explicit Euler is a simple algorithm that is easily understood and can be applied with ease by anyone with a rudimentary knowledge of computer programming. It provides reasonably accurate estimates for option prices, delta and gamma and this accuracy can be improved at the cost of increased computation.



## APPENDIX : EXPLICIT EULER VBA CODE

The tables and chart in this paper were produced using the following VBA code:

```

Public Function ExplicitEuler( _
    sglStrike As Single, _
    sglSigma As Single, _
    sglRfrate As Single, _
    sglTerm As Single, _
    intNAS As Integer, _
    Optional strType As String = "Call", _
    Optional strOutput As String _
) As Variant

Dim j As Integer, _
    k As Integer, _
    intNTS As Integer, _
    sglDeltaS As Double, _
    sglDeltaT As Single, _
    sglarrTM() As Single, _
    vararrVold As Variant, _
    vararrVnew As Variant, _
    sglarrChartData() As Single

'// ensure stability of Explicit Euler
intNAS = Application.WorksheetFunction.Max(intNAS, Int(sglRfrate / sglSigma ^ 2) + 1) '//' satisfy constraint on # asset steps
intNTS = Int((intNAS ^ 2 * sglSigma ^ 2 + sglRfrate) * sglTerm) + 1 '//' satisfy constraint on # time steps

'// determine size of steps
sglDeltaS = 2 * sglStrike / intNAS '//' note: Smax = 2 x Strike Price
sglDeltaT = sglTerm / intNTS

'// redimension Transition Matrix (TM) and vectors of option value estimates (V-old an V-new)
ReDim sglarrTM(0 To intNAS, 0 To intNAS), _
    vararrVold(0 To intNAS, 0 To 0), _
    vararrVnew(0 To intNAS, 0 To 0), _
    sglarrChartData(0 To intNAS, 0 To intNTS)

'// place FD coefficients (aj, bj and cj) into Transition Matrix
For j = 1 To intNAS - 1
    sglarrTM(j, j - 1) = j * sglDeltaT / 2 * (j * sglSigma ^ 2 - sglRfrate) '//' aj
    sglarrTM(j, j) = 1 - j ^ 2 * sglSigma ^ 2 * sglDeltaT - sglRfrate * sglDeltaT '//' bj
    sglarrTM(j, j + 1) = j * sglDeltaT / 2 * (j * sglSigma ^ 2 + sglRfrate) '//' cj
Next j

'// set boundary conditions in Transition Matrix:
'// as S->0
sglarrTM(0, 0) = 1 - sglRfrate * sglDeltaT
'// as S -> Smax
sglarrTM(intNAS, intNAS - 3) = -1 * sglarrTM(intNAS - 2, intNAS - 3)
sglarrTM(intNAS, intNAS - 2) = 2 * sglarrTM(intNAS - 1, intNAS - 2) - sglarrTM(intNAS - 2, intNAS - 2)
sglarrTM(intNAS, intNAS - 1) = 2 * sglarrTM(intNAS - 1, intNAS - 1) - sglarrTM(intNAS - 2, intNAS - 1)
sglarrTM(intNAS, intNAS) = 2 * sglarrTM(intNAS - 1, intNAS)

'// set initial condition (= option payoff at expiry) in vector of option value estimates
For j = 0 To intNAS
    vararrVold(j, 0) = Application.WorksheetFunction.Max(0, (j * sglDeltaS - sglStrike) * IIf(strType = "Put", -1, 1))
    sglarrChartData(j, intNTS) = vararrVold(j, 0)
Next j

Select Case strOutput

Case "Chart"
    '//' now step through each time step (ie. going 'back in time') to reach our answer
    For k = 1 To intNTS
        '//' WARNING: WorksheetFunction.MMult redimensions vararrVnew to Base 1
        vararrVnew = Application.WorksheetFunction.MMult(sglarrTM, vararrVold)
        vararrVold = vararrVnew
        '//' capture vararrVold values into a matrix to be used for Charting
        For j = 0 To intNAS
            sglarrChartData(j, intNTS - k) = vararrVold(j + 1, 1)
        Next j
    Next k
    ExplicitEuler = sglarrChartData '//' return NAS x NTS matrix of values for Chart
Exit Function

```

Case "Greeks"

```
'// now step through each time step (ie. going 'back in time') to reach our answer
For k = 1 To intNTS
    '// WARNING: WorksheetFunction.MMult redimensions vararrVnew to Base 1
    vararrVnew = Application.WorksheetFunction.MMult(sglarrTM, vararrVold)
    vararrVold = vararrVnew
Next k
'// add Delta and Gamma estimates to option price estimates
ReDim Preserve vararrVnew(1 To intNAS + 1, 1 To 3)
For j = 2 To intNAS
    vararrVnew(j, 2) = (vararrVnew(j + 1, 1) - vararrVnew(j - 1, 1)) / (sglDeltaS * 2) '// Delta
    vararrVnew(j, 3) = (vararrVnew(j + 1, 1) - 2 * vararrVnew(j, 1) + vararrVnew(j - 1, 1)) / (sglDeltaS ^ 2) '// Gamma
Next j
ExplicitEuler = vararrVnew '// return NAS x 3 matrix of option prices, deltas and gammas
Exit Function
```

Case Else

```
'// now step through each time step (ie. going 'back in time') to reach our answer
For k = 1 To intNTS
    '// WARNING: WorksheetFunction.MMult redimensions vararrVnew to Base 1
    vararrVnew = Application.WorksheetFunction.MMult(sglarrTM, vararrVold)
    vararrVold = vararrVnew
Next k
ExplicitEuler = vararrVold '// return single column vector of option prices
Exit Function
```

End Select

End Function

APPENDIX B: DERIVING DELTA AND GAMMA

**For a Black-Scholes Call Option**

The closed-form solution for a call option's price under Black-Scholes' assumptions is:

$$C = S \cdot N(d_1) - K \cdot e^{-r \cdot (T-t)} \cdot N(d_2)$$

where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) \cdot (T-t)}{\sigma \cdot \sqrt{T-t}} \quad \text{and} \quad d_2 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right) \cdot (T-t)}{\sigma \cdot \sqrt{T-t}} = d_1 - \sigma \cdot \sqrt{T-t}$$

therefore:

$$\begin{aligned} \frac{\partial C}{\partial S} &= S \cdot \frac{\partial N(d_1)}{\partial S} + N(d_1) - K \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial N(d_2)}{\partial S} \\ &= N(d_1) + S \cdot \frac{\partial N(d_1)}{\partial S} - K \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial N(d_2)}{\partial S} \\ &= N(d_1) + S \cdot \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{\partial d_1}{\partial S} - K \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial [N(d_2)]}{\partial d_2} \cdot \frac{\partial d_2}{\partial S} \\ &= N(d_1) + S \cdot \frac{\partial [N(d_1)]}{\partial d_1} \cdot \left[ \frac{1}{\sigma \cdot \sqrt{T-t}} \cdot \frac{\partial \ln\left(\frac{S}{K}\right)}{\partial S} \right] - K \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial [N(d_2)]}{\partial d_2} \cdot \left[ \frac{1}{\sigma \cdot \sqrt{T-t}} \cdot \frac{\partial \ln\left(\frac{S}{K}\right)}{\partial S} \right] \\ &= N(d_1) + S \cdot \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} \cdot \frac{K}{S} \cdot \frac{1}{K} - K \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial [N(d_2)]}{\partial d_2} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} \cdot \frac{K}{S} \cdot \frac{1}{K} \\ &= N(d_1) + \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} - \frac{K}{S} \cdot e^{-r \cdot (T-t)} \cdot \frac{\partial [N(d_2)]}{\partial d_2} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} \end{aligned}$$

however, since  $d_2 \sim N(0, 1)$  and  $d_2 = d_1 - \sigma \cdot \sqrt{T-t}$  its probability density functions (p.d.f) is:

$$\begin{aligned} \frac{\partial [N(d_2)]}{\partial d_2} &= \frac{e^{-\frac{1}{2} \cdot (d_2)^2}}{\sqrt{2 \cdot \pi}} \\ &= \frac{e^{-\frac{1}{2} \cdot (d_1 - \sigma \cdot \sqrt{T-t})^2}}{\sqrt{2 \cdot \pi}} \\ &= \frac{e^{-\frac{1}{2} \cdot [d_1^2 - 2 \cdot d_1 \cdot \sigma \cdot \sqrt{T-t} + \sigma^2 \cdot (T-t)]}}{\sqrt{2 \cdot \pi}} \end{aligned}$$

$$\begin{aligned}
 &= \frac{e^{-\frac{1}{2} \cdot (d_1)^2} \cdot e^{d_1 \cdot \sigma \cdot \sqrt{T-t}} \cdot e^{-\frac{\sigma^2}{2} \cdot (T-t)}}{\sqrt{2 \cdot \pi}} \\
 &= \frac{e^{-\frac{1}{2} \cdot (d_1)^2} \cdot e^{\ln\left(\frac{S}{K}\right) + r \cdot (T-t) + \frac{\sigma^2}{2} \cdot (T-t)} \cdot e^{-\frac{\sigma^2}{2} \cdot (T-t)}}{\sqrt{2 \cdot \pi}} \\
 &= \frac{e^{-\frac{1}{2} \cdot (d_1)^2} \cdot e^{\ln\left(\frac{S}{K}\right) + r \cdot (T-t)}}{\sqrt{2 \cdot \pi}} \\
 &= e^{\ln\left(\frac{S}{K}\right)} \cdot e^{r \cdot (T-t)} \cdot \frac{e^{-\frac{1}{2} \cdot (d_1)^2}}{\sqrt{2 \cdot \pi}} \\
 &= \frac{S}{K} \cdot e^{r \cdot (T-t)} \cdot \frac{e^{-\frac{1}{2} \cdot (d_1)^2}}{\sqrt{2 \cdot \pi}}
 \end{aligned}$$

therefore:

$$\frac{\partial [N(d_2)]}{\partial d_2} = \frac{S}{K} \cdot e^{r \cdot (T-t)} \cdot \frac{\partial [N(d_1)]}{\partial d_1}$$

and hence:

$$\begin{aligned}
 \frac{\partial C}{\partial S} &= N(d_1) + \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} - \frac{K}{S} \cdot e^{-r \cdot (T-t)} \cdot \frac{S}{K} \cdot e^{r \cdot (T-t)} \cdot \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} \\
 &= N(d_1) + \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} - \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{\sigma \cdot \sqrt{T-t}} \\
 &= N(d_1)
 \end{aligned}$$

ie.

$$\frac{\partial C}{\partial S} = N(d_1)$$

hence:

$$\begin{aligned}
 \frac{\partial^2 C}{\partial S^2} &= \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{S \cdot \sigma \cdot \sqrt{T-t}} \\
 &= \frac{e^{-\frac{1}{2} \cdot (d_1)^2}}{\sqrt{2 \cdot \pi}} \cdot \frac{1}{S \cdot \sigma \cdot \sqrt{T-t}}
 \end{aligned}$$

### For a Black-Scholes Put

The formula for a put is similar to a call but with a few signs reversed:

$$P = -S \cdot N(-d_1) + K \cdot e^{-r(T-t)} \cdot N(-d_2)$$

where  $d_1$  and  $d_2$  are as before. The delta, therefore, is:

$$\frac{\partial P}{\partial S} = -S \cdot \frac{\partial N(-d_1)}{\partial S} - N(-d_1) + K \cdot e^{-r(T-t)} \cdot \frac{\partial N(-d_2)}{\partial S}$$

Now recall the symmetry of the Normal probability density function:  $N(-d_1) = 1 - N(d_1)$ . Hence:

$$\begin{aligned} \frac{\partial P}{\partial S} &= -S \cdot \frac{\partial [1 - N(d_1)]}{\partial S} - [1 - N(d_1)] + K \cdot e^{-r(T-t)} \cdot \frac{\partial [1 - N(d_2)]}{\partial S} \\ &= S \cdot \frac{\partial N(d_1)}{\partial S} - 1 + N(d_1) - K \cdot e^{-r(T-t)} \cdot \frac{\partial N(d_2)}{\partial S} \\ &= N(d_1) - 1 + \left[ S \cdot \frac{\partial N(d_1)}{\partial S} - K \cdot e^{-r(T-t)} \cdot \frac{\partial N(d_2)}{\partial S} \right] \end{aligned}$$

which, from the call option above, we recognise reduces to:

$$\frac{\partial P}{\partial S} = N(d_1) - 1$$

and

$$\begin{aligned} \frac{\partial^2 P}{\partial S^2} &= \frac{\partial [N(d_1)]}{\partial d_1} \cdot \frac{1}{S \cdot \sigma \cdot \sqrt{T-t}} \\ &= \frac{e^{-\frac{1}{2} \cdot (d_1)^2}}{\sqrt{2 \cdot \pi}} \cdot \frac{1}{S \cdot \sigma \cdot \sqrt{T-t}} \\ &= \frac{\partial^2 C}{\partial S^2} \end{aligned}$$