University of Richmond
UR Scholarship Repository

Finance Faculty Publications

Finance

2024

Black-Scholes Option Pricing and Greeks Using Excel's "LAMBDA" Function

Tom Arnold
University of Richmond, tarnold@richmond.edu

Joseph Farizo
University of Richmond, jfarizo@richmond.edu

Jonathan M. Godbey

Follow this and additional works at: https://scholarship.richmond.edu/finance-faculty-publications

Part of the Finance and Financial Management Commons

Recommended Citation

Arnold, Tom, C., Joseph Farizo, Jonathan M. Goodbey. "Black-Scholes Option Pricing and Greeks Using Excel's "LAMBDA" Function," University of Richmond Robins School of Business, (2024): 1-10.

This Working Paper is brought to you for free and open access by the Finance at UR Scholarship Repository. It has been accepted for inclusion in Finance Faculty Publications by an authorized administrator of UR Scholarship Repository. For more information, please contact scholarshiprepository@richmond.edu.

Black-Scholes Option Pricing and Greeks Using Excel's "LAMBDA" Function

Tom Arnold, CFA, CIPM
The Robins School of Business
University of Richmond
Department of Finance
102 UR Drive
Richmond, VA 23173
tarnold@richmond.edu
O: 804-287-6399
F: 804-289-8878

Joseph Farizo
The Robins School of Business
Department of Finance
102 UR Drive
University of Richmond, VA 23173
O: 804-289-8565
F: 804-289-8878
jfarizo@richmond.edu

Jonathan M. Godbey
Robinson College of Business
Georgia State University
Department of Finance
33 Gilmer Street SE
Atlanta, GA 30303
Oxford, OH 45056-1879
jgodbey@gsu.edu
O: 404-413-7328
F: 404-413-7312

Please do not cite without permission

August 20, 2024

Black-Scholes Option Pricing and Greeks Using Excel's "LAMBDA" Function

The =LAMBDA function within Excel provides a powerful new tool for investors and analysts. In this treatment, we show how to create a function that calculates an option's intrinsic value, price, and delta based on the Black-Scholes model. Other option Greek functions and calculations are available in a downloadable file. The LAMBDA function is not limited to the Black-Scholes model and has important advantages over Excel's previous solution of creating user-defined functions in VBA.

INTRODUCTION

The Black-Scholes option pricing model (1973) is still a fundamental part of option pricing and, to some extent, the basis to which more complex models are compared. If the complexity of a model is not sufficiently more accurate than the Black-Scholes model, the extra complexity is not pragmatically justifiable. Further, partial derivatives of the Black-Scholes model, i.e. the "Greeks", are still important for hedging and determining the sensitivity of a portfolio containing options relative to underlying security price changes, volatility changes, etc. For this reason, there is a benefit to creating Excel functions that require a standard set of parameters to produce Black-Scholes option prices and Greeks.

Excel's new "=LAMBDA" function allows for the creation of Black-Scholes option pricing and Greeks in the form of a "named" function that becomes available in the function menu. If one copies a cell containing one of these newly created functions, and "pastes" the cell into a second Excel file, the newly created function becomes available in the function menu in the second Excel file. In other words, once created the function is easily transferable to other Excel files. We recommend that users maintain a spreadsheet with a comprehensive list of LAMBDA functions that they have created, so that these functions may be copied into newly created spreadsheets as needed.

Microsoft announced the addition of the LAMBDA function on December 3, 2020: https://insider.microsoft365.com/en-us/blog/lambda-excel-custom-functions. Microsoft's goal was to allow users to create their own functions without having to learn how to create VBA User Defined Functions. Also, =LAMBDA allows users to define their own functions without saving files as macro-enabled.

Essentially, =LAMBDA function is a generic format for an Excel function:

=LAMBDA(parameter1, parameter2,..parameterN,calculation based on listed parameters)

However, a "name" can be created for the function, for example "XYZ." The function =XYZ can now be implemented anywhere within the Excel file as:

=XYZ(parameter1, parameter2,..parameterN)

The =XYZ function will then produce the calculation within the associated =LAMBDA function from the parameters (usually cell addresses) designated within the =XYZ function. Further, the =XYZ function and standard Excel functions can be nested within other newly created functions. For example, a user defined function can be created for the Black-Scholes variable "d_1", which will then be used within other user defined functions to price a call option and to find option deltas.

In the next section, we demonstrate the implementation of =LAMBDA function to create a "named" user defined function and apply the function to price options and option deltas. Section two provides more applications of creating user defined functions that incorporate a cost of capital. The third section concludes.

SECTION 1: =LAMBDA Functions and Black-Scholes Option Pricing Applications

Define the following variables: underlying security spot price (S), exercise price (X), underlying security return volatility (V), days to maturity (DTM), days in a year (DIY), and the risk-free security return (RF). The Black-Scholes Option Pricing Model (1973) values a call option (C) as:

$$C = S \times N(d_1) - X \times EXP(-RF \times DTM / DIY) \times N(d_2) \quad (1)$$

$$d_1 = [LN(S / X) + (RF + V^2 / 2) \times (DTM / DIY)] / [V \times (DTM / DIY)^{1/2}] \quad (2)$$

$$d_2 = d_1 - [V \times (DTM / DIY)^{1/2}] \quad (3)$$

where, N(*) is the cumulative normal distribution, LN(*) is the natural logarithm, and EXP(*) is the exponential function

Similarly, a put option (P) is valued as:

$$P = X \times EXP(-RF \times DTM / DIY) \times N(-d_2) - S \times N(-d_1) \quad (4)$$

Or, based on put-call parity:

$$P = C + X \times EXP(-RF \times DTM / DIY) - S \quad (5)$$

To introduce the process for creating a "named" function using the =LAMBDA function, a simpler calculation is performed based on an option's intrinsic value. For a call option, the intrinsic value is the MAX((S – X), 0), i.e., the maximum of (S – X) and zero. For a put option, the intrinsic value is MAX ((X – S), 0), i.e. the maximum of (X – S) and zero. To create a new function " =INTRINSICC", that determines the intrinsic value of a call option:

1. Double-click the "Name manager" icon in the "Formulas" tab
2. Click the "New" button and the following menu appears:
    a. Name: set as INTRINSICC (i.e. the name of the function)
    b. Scope: set to "Workbook" as the default
    c. Comment: write description of the function (optional)
    d. Refers to: set as =LAMBDA(S,X,MAX(S – X, 0))
3. Click the "OK" button
4. Click the "Close" button

As stated above, =LAMBDA is the template for the =INTRINSICC function. There are two parameters: S and X, and a calculation, MAX(S – X, 0), based on the parameters. After performing the four steps, a new user defined "named" function is available: =INTRINSICC(S,X) that returns the MAX(S – X, 0) based on the parameters S and X. The =INTRINSICC function is also now available within the function menu in Excel.

To create the associated put option intrinsic value function =INTRINSICP, follow the same steps with Name set as INTRINSICP, and Refers to: =LAMBDA(S,X,MAX(X – S,0)). Again, a new user defined "named" function becomes available: =INTRINSICP(S,X) that returns the MAX(X – S, 0) based on the parameters S and X.

Figure 1 displays newly created functions with values provided for S, X, V, DTM, DIY, and RF based on cell addresses:

- =INTRINSICC(S,X), calculates the intrinsic value of a call option
- =INTRINSICP(S,X), calculates the intrinsic value of a put option
- =D1BS(S,X,V,DTM,DIY,RF), calculates "d_1"
- =D2BS(S,X,V,DTM,DIY,RF), calculates "d_2"
- =BSCALL(S,X,V,DTM,DIY,RF), calculates the value of a call option
- =BSPUT(S,X,V,DTM,DIY,RF), calculates the value of a put option
- =DELTAC(S,X,V,DTM,DIY,RF), calculates the "delta" of a call option
- =DELTAP(S,X,V,DTM,DIY,RF), calculate the "delta" of a put option

Figure 1: =LAMBDA Functions for Call and Put Options

	A	B	C	D	E	F	G
1	**Option Information:**						
2							
3	Stock Price:	$ 26.00	S		Cost of Carry:		Q
4	Strike Price:	$ 25.00	X				
5	Volatility:	46.00%	V				
6	Days to Maturity:	130	DTM				
7	Days in a Year:	360	DIY				
8	Risk-free Rate:	2.20%	RF				
9							
10	Intrinsic Value-Call:	$ 1.00					
11	Intrinsic Value-Put:	$ -					
12					**ADJUSTED for Q:**		
13	d1:	0.308838			d1:		
14	d2:	0.032412			d2:		
15	BS-Call:	$ 3.431477			BS-Call:		
16	BS-Put:	$ 2.233653			BS-Put:		
17	Delta-Call:	0.621278			Delta-Call:		

18	Delta-Put:	-0.378722			Delta-Put:		
19							

Cell B10: =INTRINSICC(B3, B4), created from:
=LAMBDA(S,X,MAX(S – X, 0))

Cell B11: =INTRINSICP(B3, B4), created from:
=LAMBDA(S,X,MAX(X – S, 0))

Cell B13: =D1BS(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,(LN(S/X) +(RF+V*V/2)*(DTM/DIY))/(V*SQRT(DTM/DIY)))

Cell B14: =D2BS(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,D1BS(S,X,V,DTM,DIY,RF) – V*SQRT(DTM/DIY))

Cell B15: =BSCALL(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,S*NORMSDIST(D1BS(S,X,V,DTM,DIY,RF)) – X*EXP(-RF*DTM/DIY)
*NORMSDIST(D2BS(S,X,V,DTM,DIY,RF)))

Cell B16: =BSPUT(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF, X*EXP(-RF*DTM/DIY)*NORMSDIST(-D2BS(S,X,V,DTM,DIY,RF)) – S
*NORMSDIST(-D1BS(S,X,V,DTM,DIY,RF)))

Cell B17: =DELTAC(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,NORMSDIST(D1BS(S,X,V,DTM,DIY,RF)))

Cell B18: =DELTAP(B3, B4, B5, B6, B7, B8) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,NORMSDIST(D1BS(S,X,V,DTM,DIY,RF)) – 1)

A copy of this spreadsheet is available at: https://scholarship.richmond.edu/finance-faculty-publications/XX/

As with the =INTRINSICC and =INTRINSICP functions, Figure 1 provides the function name (without the equal sign) to be used and the associated =LAMBDA function to define the new "named" user defined function.

There are few items to note in Figure 1. First, notice "standard" Excel functions: =NORMSDIST, =LN, and =EXP can be called within a =LAMBDA function. Further, by creating functions =D1BS and =D2BS for the calculation of "d_1" and "d_2", these functions are nested within the functions: =BSCALL, =BSPUT, =DELTAC, and =DELTAP. Even if cells B13 and B14 are deleted, the four functions for option pricing and option deltas will still function. This is important from a programming perspective, in which intermediate calculations can be performed for a final result as a set of =LAMBDA functions. Then, only the final result function will need to be incorporated into the spreadsheet. It is in this manner that =LAMBDA functions can substitute for calculations that used to require VBA (Visual Basic Application) programming.

The option deltas are the respective partial derivatives of a call and put option relative to the underlying security price (S).

$$\text{DELTA(C)} = \text{N}(d_1) \qquad (6)$$

$$\text{DELTA(P)} = \text{N}(d_1) - 1 \qquad (7)$$

The delta calculations are provided by =DELTAC for a call option and =DELTAP for a put option in Figure 1. Other Greeks are available for options with respect to the second partial derivative relative to S (gamma), the first partial derivative relative to T (T = DTM / DIY, theta), the first patrial derivative relative to V (vega), and the first partial derivative relative to RF (rho). These additional Greeks are not presented here, but are available as functions in a spreadsheet that is downloadable at:

https://scholarship.richmond.edu/finance-faculty-publications/XX/

SECTION 2: =LAMBDA Functions for Options with a Cost of Carry

Based on Hull and Basu (2022, Table 19.6), we introduce the cost of carry to the Black-Scholes calculations in Figure 1. The cost of carry (Q) is set at 5.00% APR and the associated calculations are provide in Figure 2:

Figure 2: =LAMBDA Functions for Call and Put Options with a Cost of Carry

	A	B	C	D	E	F	G
1	**Option Information:**						
2							
3	Stock Price:	$ 26.00	S		Cost of Carry:	5.00%	Q
4	Strike Price:	$ 25.00	X				
5	Volatility:	46.00%	V				
6	Days to Maturity:	130	DTM				
7	Days in a Year:	360	DIY				
8	Risk-free Rate:	2.20%	RF				
9							

10	Intrinsic Value-Call:	$ 1.00					
11	Intrinsic Value-Put:	$ -					
12					**ADJUSTED for Q:**		
13	d1:	0.308838			d1:	0.243520	
14	d2:	0.032412			d2:	-0.032906	
15	BS-Call:	$ 3.431477			BS-Call:	$3.148238	
16	BS-Put:	$ 2.233653			BS-Put:	$2.415646	
17	Delta-Call:	0.621278			Delta-Call:	0.585531	
18	Delta-Put:	-0.378722			Delta-Put:	-0.396576	
19							

Cell F13: =D1BSQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,D1BS(S,X,V,DTM,DIY,(RF – Q)))

Cell F14: =D2BSQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,D2BS(S,X,V,DTM,DIY,(RF – Q)))

Cell F15: =BSCALLQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,S*EXP(-Q*DTM/DIY)*NORMSDIST(D1BSQ(S,X,V,DTM,DIY,RF,Q))
– X*EXP(-RF*DTM/DIY)*NORMSDIST(D2BSQ(S,X,V,DTM,DIY,RF,Q)))

Cell F16: =BSPUTQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,X*EXP(-RF*DTM/DIY)*NORMSDIST(-
D2BSQ(S,X,V,DTM,DIY,RF,Q)) – S*EXP(-Q*DTM/DIY)*NORMSDIST(-D1BSQ(S,X,V,DTM,DIY,RF,Q)))

Cell F17: =DELTACQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,NORMSDIST(D1BSQ(S,X,V,DTM,DIY,RF,Q))*EXP(-Q*DTM/DIY))

Cell F18: =DELTAPQ(B3, B4, B5, B6, B7, B8,F3) created from:
=LAMBDA(S,X,V,DTM,DIY,RF,Q,NORMSDIST(D1BSQ(S,X,V,DTM,DIY,RF,Q)) – 1)*EXP(-
Q*DTM/DIY))

A copy of this spreadsheet is available at: https://scholarship.richmond.edu/finance-faculty-publications/XX/

Each cost of carry function has a similar name to the associated function in Figure 1, but ends with the letter "Q". The cost of carry functions also use named functions from Figure 1 and have functions nested within functions, similar to Figure 1. Again, additional Greek calculations/functions with the cost of carry: gamma, theta, vega, and rho, are available as functions in a spreadsheet that is downloadable at:

https://scholarship.richmond.edu/finance-faculty-publications/XX/

SECTION 3: CONCLUSION

The =LAMBDA function within Excel provides a powerful new tool for investors and analysts. Based on the Black-Scholes model, we create user defined named functions to calculate an option's intrinsic value, price, and delta with and without a cost of carry component. Other option Greek functions are available in a downloadable file.

One should not feel limited to only the Black-Scholes model. More complex option pricing models or other security pricing models can be implemented using Excel's =LAMBDA function. If the model requires intermediate calculations to produce a final calculation, each intermediate calculation can be defined as a named function that are then implemented (or nested) within the named function for the final result. Only the "final result function" needs to exist within the spreadsheet.

For example, =BSCALL can be used within a spreadsheet without having to also have cells with =D1BS and =D2BS calculated within the same spreadsheet. Further, if one copied the =BSCALL function from a cell within one spreadsheet to a second spreadsheet, =BSCALL, =D1BS, and =D2BS will be available in the second spreadsheet because the latter two functions are nested within the first function. Consequently, it is very easy to transfer user defined functions from one file to another and to have multiple step processes within one user defined function.

REFERENCES

Black, Fischer and Myron Scholes, 1973. “The Pricing of Options and Corporate Liabilities.” *Journal of Political Economy*, V. 81, N. 3 pp. 637 – 654.

Hull, John C. and Sankarshan Basu. 2022. Options, Futures, and Other Derivative Securities (11th Edition, Indian Edition), Pearson Education Inc.

https://insider.microsoft365.com/en-us/blog/lambda-excel-custom-functions