Project Overview:

The application is an option valuation program with a GUI front-end. The application is used to model the expected price of an option financial instrument.

Most derivatives (like forwards, futures, and swaps) have prices quoted daily by brokers and exchanges, which makes price discovery somewhat straight forward. Options, however, use a model to determine current estimated market price. What would be helpful would be an application that can be used to model option pricing based on key inputs (noted below) while also allowing the user to alter some of the key inputs (say, volatility) to better understand how that would impact the option valuation – my application does just this.

This application may not be flashy but that does not mean that it is not very practical or simple (the underlying valuation engine is non-trivial).

Fig.1 – initial screen

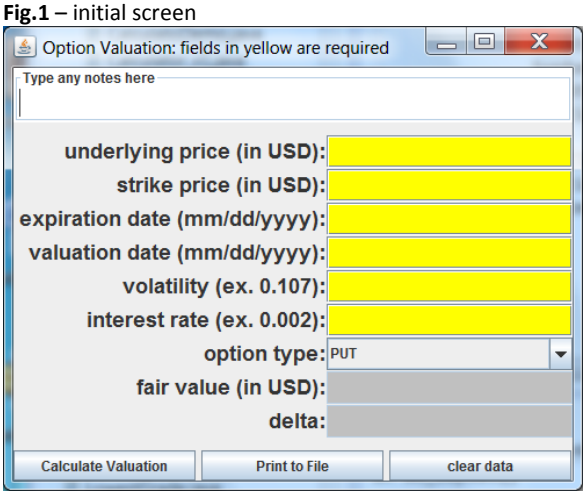
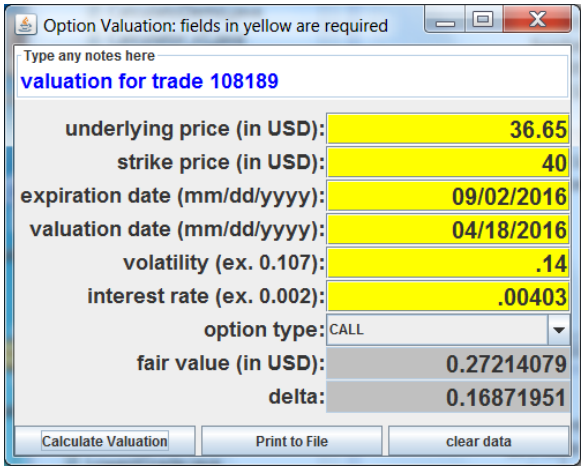


Fig.2 – with example data and results



Need for Application:

Financial instruments like options are traded daily to allow companies a way to hedge or to speculate. As mentioned, valuing an option is not straight-forward and tools have been developed to assist with valuation. Unfortunately, most of these tools are quite expensive. The application presented in this final project will allow individuals a free way to value options and perform various "what if" type scenarios ("What if volatility doubled?", "What if interest rates fell?", "What if time to maturity were increased?")

Instructions:
The GUI is loaded by executing the *TermProject_Blackford.java* file. At start-up the user will be presented with the GUI noted at **Fig. 1** above. All fields in yellow are required and the GUI will provide pop-up messages reminding the user to complete necessary fields. Fields entered improperly, such as dates, will also cause a pop-up to allow the use to correct input.

The *option type* input is made from a drop down, where the user selects if the option is a PUT or a CALL.

The *notes* field allows optional text input to describe the valuation. Helpful information could be the transaction number, the type of option (is it a commodity like coal, or natural gas), or various "what if" type scenarios like changes to volatility to determine valuation impact.

Buttons:

- **Create Valuation** – this button will populate the *fair value* (option valuation) and the *delta*. If required fields are missing pop-up messages will alert the user.

- **Print to File** – this button will output the valuation results to a pipe ("|") delimited text file named ValuationFile.txt. As a pipe delimited file it is easily read by Excel (see example output at **Exhibit C**). The output file will continue to add new rows as additional valuations are made, with each new row date and time stamped.

- **Clear Data** – this button will clear all the data fields as a convenience for entry of new option data.

Features:
When proposed the project was only to perform the option valuation. Final implementation also contains the calculated delta. After the valuation the delta is often the key calculation used by traders to determine when to buy or sell the instrument.

The application will also output results to a .txt file which is easily imported into Excel (see **Print to File** button information above).

Main Algorithm:

Options are valued based on the model developed by Merton, Scholes and Black, generally refer to as the "Black-Scholes" model. The model is based on the following (per Wikipedia[1]):

The value of a call option for a non-dividend-paying underlying stock in terms of the Black–Scholes parameters is:

$$\begin{aligned} C(S,t) &= N(d_1)S - N(d_2)Ke^{-r(T-t)} \\ d_1 &= \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right] \\ d_2 &= d_1 - \sigma\sqrt{T-t} \end{aligned}$$

The price of a corresponding put option based on put–call parity is:

$$\begin{aligned} P(S,t) &= Ke^{-r(T-t)} - S + C(S,t) \\ &= N(-d_2)Ke^{-r(T-t)} - N(-d_1)S \end{aligned}$$

For both, as above:

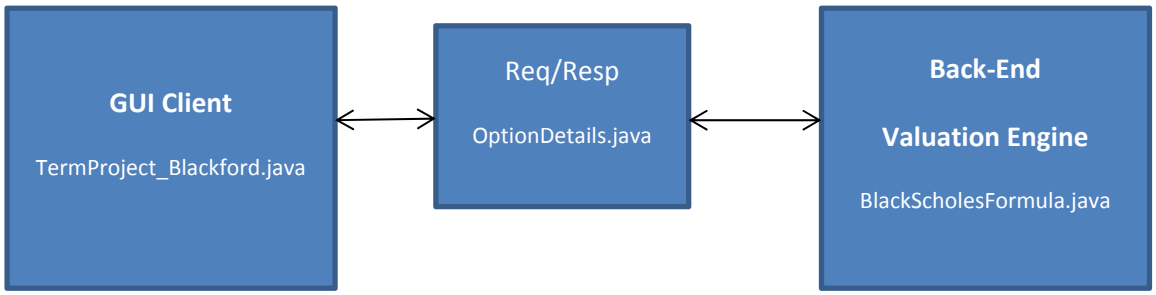
- $N(\cdot)$ is the cumulative distribution function of the standard normal distribution
- $T-t$ is the time to maturity
- S is the spot price of the underlying asset
- K is the strike price
- r is the risk free rate (annual rate, expressed in terms of continuous compounding)
- σ is the volatility of returns of the underlying asset

Although not trivial the model is well known and extensively used. Even though the model equation is publicly available it is often implemented in fee-based tools (like FINCAD) and developing a free GUI interface would be immediately helpful in my work.

[1] "Black–Scholes model" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 22nd July 2004. Web. 4 April 2016.
<https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model>

Expansion:
Part of an option valuation is the calculation of what are known as "the Greeks": delta, gamma, theta, vega, and rho. Delta is included in the current version of the application but it would be nice to also implement the remaining Greeks.

Code Specifics:
The application has two key distinct components: the GUI front end and the back-end valuation model.



GUI Client – built utilizing standard Java swing and is architected to be separate from the underlying option calculations. This client implementation is designed to provide a clean user interface for entry of required request fields and the display of results. See example at **Fig. 1**.

Back-End Valuation Engine – built utilizing Java and is architected to calculate the option valuation based on received input. The interface is very basic, intended to be received from a client.

Request / Response - this is a simple class intended to pass data to/from the front-end client and the back-end valuation engine. My preference is not to utilize getters and setters due to the nature of the class, but to keep with the conventions of CSCI e-10b I added getters and setters.

Reactions:
I am quite pleased with the current version of the project. Already the application has found frequent use in my daily work, assisting with the quick review of various option scenarios. The application may not be flashy but it is practical. What superficially appears to be a simple tool involved a large number of programming hours. Simple concepts can sometime take a surprising amount of time to bring to completion.

The back-end server took a while to test to ensure appropriate valuation results. I wanted to use only standard Java classes instead of some financial math packages made available on-line – this added to development time. Testing was done by comparing results with known results (see **exhibit A** and **exhibit B**).

Working with Java dates and times was also surprisingly difficult. Attractive easy to read file output of dates and times is difficult to format.

Overall I am quite pleased and proud of the end result. The application serves a very narrow user group but is very helpful to that demographic.

Exhibit A: valuation of options using FinCAD[2]

trade #	Underlying price	Strike	Expiration Dt	valuation dt	volatility	interest	type CALL=1 PUT =2	stat	accrual	valuation	delta
107412	8.9	13	9/2/2016	4/18/2016	0.107	0.00403	1	1	1	0.00000	0.00000
108034	8.87	9.75	6/2/2016	4/18/2016	0.107	0.00289	2	1	1	0.880349199	(0.99342)
108047	8.87	9.75	6/2/2016	4/18/2016	0.107	0.00289	2	1	1	0.880349199	(0.99342)
108156	0.77	0.74	6/30/2016	4/18/2016	0.126	0.00308	2	1	1	0.006018429	(0.23151)
108167	0.7689	0.75	7/29/2016	4/18/2016	0.12713	0.00339	2	1	1	0.012275715	(0.34281)
108180	42.95	36	12/2/2016	4/18/2016	0.21141	0.00534	2	1	1	0.488323116	(0.12673)
108188	45.75	33	9/2/2016	4/18/2016	0.2025	0.00403	2	1	1	0.006323934	(0.00351)
108189	36.65	40	9/2/2016	4/18/2016	0.14	0.00403	1	1	1	0.262917507	0.16407
108218	1.997	2.05	5/26/2016	4/18/2016	0.4409	0.00285	1	1	1	0.090153924	0.45488

Above option valuations were made using FinCAD, an expensive Excel add-in. Calculations used to verify results of project (last two columns are the results which compare to project results)

Exhibit B: valuation of options using Allegro[3]

TRADE	EXECUTION	X_SUNDERLYING	STRIKE PRICE	EXPIRATION DATE	VALUATION DATE	X_VOLATILITY	X_INTEREST	OPTION TYPE	MARKET PRICE	DELTA
107412	Coal Option Financial	$8.9000	$13.00	9/2/2016	5/2/2016	0.10700	0.00403	CALL	$0.0000	0.00000
108034	Coal Option Financial	$8.8700	$9.75	6/2/2016	5/2/2016	0.10700	0.00289	PUT	$0.8803	(0.99342)
108047	Coal Option Financial	$8.8700	$9.75	6/2/2016	5/2/2016	0.10700	0.00289	PUT	$0.8803	(0.99342)
108156	FX Option Financial	$0.7700	$0.74	6/30/2016	5/2/2016	0.12600	0.00308	PUT	$0.0060	(0.23151)
108167	FX Option Financial	$0.7689	$0.75	7/29/2016	5/2/2016	0.12713	0.00339	PUT	$0.0123	(0.34281)
108180	Coal Option Financial	$42.9500	$36.00	12/2/2016	5/2/2016	0.21141	0.00534	PUT	$0.4883	(0.12672)
108188	Coal Option Financial	$45.7500	$33.00	9/2/2016	5/2/2016	0.20250	0.00403	PUT	$0.0063	(0.00351)
108189	Coal Option Financial	$36.6500	$40.00	9/2/2016	5/2/2016	0.14000	0.00403	CALL	$0.2629	0.16407
108218	NG Option Financial	$1.9970	$2.05	5/26/2016	5/2/2016	0.44090	0.00285	CALL	$0.0912	0.45581

Above option valuations were made using Allegro, an expensive deal capture and portfolio valuation application. Calculations used to verify results of project (last two columns are the results which compare to project results)

[2] About FinCAD <http://www.fincad.com/about-fincad>

[3] About Allegro <http://www.AllegroDev.com>

Exhibit C: valuation of options using application (ValuationFile.txt)

processing date	underlying price	strike price	volatility	intRate	expiration date	valuation date	timeToExpiration	option type	valuation	delta	notes
Thu May 05 07:30:16 CDT 2016	$8.90	$13.00	0.107	0.00403	Fri Sep 02 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.37534	CALL	$0.00000	0.000000	trade 107412
Thu May 05 07:31:08 CDT 2016	$8.87	$9.75	0.107	0.00289	Thu Jun 02 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.12329	PUT	$0.87721	(0.993604)	trade 108034
Thu May 05 07:31:39 CDT 2016	$8.87	$9.75	0.107	0.00289	Thu Jun 02 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.12329	PUT	$0.87721	(0.993604)	trade 108047
Thu May 05 07:32:37 CDT 2016	$0.77	$0.74	0.126	0.00308	Thu Jun 30 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.20000	PUT	$0.00591	(0.228329)	trade 108156
Thu May 05 07:33:25 CDT 2016	$0.77	$0.75	0.12713	0.00339	Fri Jul 29 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.27945	PUT	$0.01203	(0.337965)	trade 108167
Thu May 05 07:34:11 CDT 2016	$42.95	$36.00	0.21141	0.00534	Fri Dec 02 00:00:00 CST 2016	Mon Apr 18 00:00:00 CDT 2016	0.62477	PUT	$0.47056	(0.123054)	trade 108180
Thu May 05 07:34:50 CDT 2016	$45.75	$33.00	0.2025	0.00403	Fri Sep 02 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.37534	PUT	$0.00609	(0.003390)	trade 108188
Thu May 05 07:35:30 CDT 2016	$36.65	$40.00	0.14	0.00403	Fri Sep 02 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.37534	CALL	$0.27214	0.168720	trade 108189
Thu May 05 07:36:09 CDT 2016	$2.00	$2.05	0.4409	0.00285	Thu May 26 00:00:00 CDT 2016	Mon Apr 18 00:00:00 CDT 2016	0.10411	CALL	$0.09042	0.455844	trade 108218