FINM3405 Derivatives and Risk Management

Week 5: The Black-Scholes European option pricing model

Dr Godfrey Smith

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

August 17, 2024

Contents

Introduction

Last week we covered the basics of options, including payoffs and pricing bounds. We also took a look at the major contracts traded on exchanges, and observed that equity options, particularly index options, are very popular. This week we start on options pricing, namely calculating the option premium which is paid by the taker to the writer. Option pricing takes up a lot of space in quantitative finance, both in academia and industry, and this week we take the first step into this vast area by considering the Black-Scholes European option pricing model. We also look at the "Greeks", which express the sensitivity of the option price to the various parameters inputs in the Black-Scholes pricing equations.

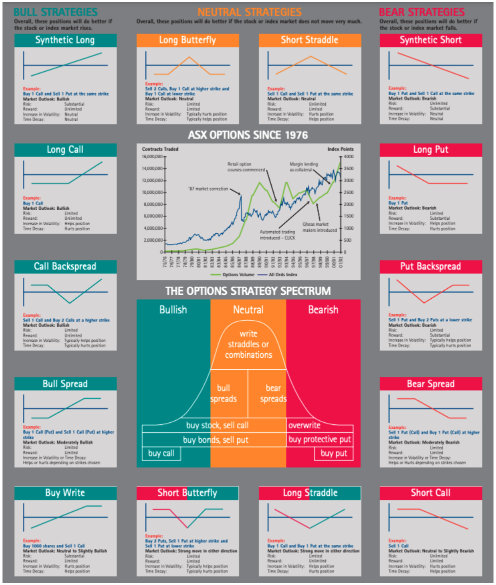
- Readings: Chapters 15 and 19 of Hull.

Introduction

There's a number of additional useful resources that I recommend considering over the 5 weeks that we cover options, including:

- The CME Group's Introduction to Options video series.
- The ASX's online Options Masterclass: Introduction to Options and their Advancing in Options Course.
- The ASX's educational documents Understanding Options Trading and Options Strategies.
- The ASX also has a cool wall poster (see next slide) that I recommend printing out and bluetacking to the wall of your study!

Introduction



BULL STRATEGIES
Overall, these positions will do better if the stock or index market rises.

NEUTRAL STRATEGIES
Overall, these positions will do better if the stock or index market does not move very much.

BEAR STRATEGIES
Overall, these positions will do better if the stock or index market falls.

Synthetic Long
Example:
Buy 1 Call and Sell 1 Put at the same strike
Market Outlook: Bullish

Risk:	Substantial
Reward:	Unlimited
Increase in Volatility:	Neutral
Time Decay:	Neutral

Long Butterfly
Example:
Sell 2 Calls, Buy 1 Call at higher strike and Buy 1 Call at lower strike
Market Outlook: Neutral

Risk:	Limited
Reward:	Limited
Increase in Volatility:	Typically hurts position
Time Decay:	Typically helps position

Short Straddle
Example:
Sell 1 Call and Sell 1 Put at the same strike
Market Outlook: Neutral

Risk:	Unlimited
Reward:	Limited
Increase in Volatility:	Hurts position
Time Decay:	Helps position

Synthetic Short
Example:
Buy 1 Put and Sell 1 Call at the same strike
Market Outlook: Bearish

Risk:	Unlimited
Reward:	Substantial
Increase in Volatility:	Neutral
Time Decay:	Neutral

Long Call
Example:
Buy 1 Call
Market Outlook: Bullish

Risk:	Limited
Reward:	Unlimited
Increase in Volatility:	Helps position
Time Decay:	Hurts position

Long Put
Example:
Buy 1 Put
Market Outlook: Bearish

Risk:	Limited
Reward:	Substantial
Increase in Volatility:	Helps position
Time Decay:	Hurts position

Call Backspread
Example:
Sell 1 Call and Buy 2 Calls at a higher strike
Market Outlook: Bullish

Risk:	Limited
Reward:	Unlimited
Increase in Volatility:	Typically helps position
Time Decay:	Typically hurts position

Put Backspread
Example:
Sell 1 Put and Buy 2 Puts at a lower strike
Market Outlook: Bearish

Risk:	Limited
Reward:	Substantial
Increase in Volatility:	Typically helps position
Time Decay:	Typically hurts position

Bull Spread
Example:
Buy 1 Call (Put) and Sell 1 Call (Put) at higher strike
Market Outlook: Moderately Bullish

Risk:	Limited
Reward:	Limited

Increase in Volatility or Time Decay:
Helps or hurts depending on strikes chosen

Bear Spread
Example:
Sell 1 Put (Call) and Buy 1 Put (Call) at higher strike
Market Outlook: Moderately Bearish

Risk:	Limited
Reward:	Limited

Increase in Volatility or Time Decay:
Helps or Hurts depending on strikes chosen

Buy Write
Example:
Buy 1000 shares and Sell 1 Call
Market Outlook: Neutral to Slightly Bullish

Risk:	Substantial
Reward:	Limited
Increase in Volatility:	Hurts position
Time Decay:	Helps position

Short Butterfly
Example:
Buy 2 Puts, Sell 1 Put at higher strike and Sell 1 Put at lower strike
Market Outlook: Strong move in either direction

Risk:	Limited
Reward:	Limited
Increase in Volatility:	Typically helps position
Time Decay:	Typically hurts position

Long Straddle
Example:
Buy 1 Call and Buy 1 Put at the same strike
Market Outlook: Strong move in either direction

Risk:	Limited
Reward:	Unlimited
Increase in Volatility:	Helps position
Time Decay:	Hurts position

Short Call
Example:
Sell 1 Call
Market Outlook: Neutral to Slightly Bearish

Risk:	Unlimited
Reward:	Limited
Increase in Volatility:	Hurts position
Time Decay:	Helps position

Introduction

The Black-Scholes European option pricing model is a mathematical equation for pricing plain vanilla European call and put options.

> In general the "Black-Scholes model" is a mathematical framework in which we derive mathematical models for pricing derivatives.

In the Black-Scholes framework, some types of derivatives have nice equations for their price or value, but most do not. When they don't, we have to use numerical methods to approximate the solution, such as:

- Lattice based method like the binomial and trinomial models.
- Monte Carlo simulation methods.
- Numerically solving partial differential equations.

Introduction

This week we cover the Black-Scholes option pricing model for:

- The basic equations for European options.
- European options on equities or indices that pay a dividend yield q.
- European currency options, being mindful of the domestic r_{d} and foreign r_{f} risk-free rates, as well as currency quoting conventions.

In later weeks we cover numerical methods for other types of options that don't have neat Black-Scholes pricing equations.

Black-Scholes model

Quite simply, the **Black-Scholes** European call option pricing model is

$$\boxed{C = S\mathcal{N}(d_1) - Ke^{-rT}\mathcal{N}(d_2),}$$

where

$$d_1 = \frac{\log \frac{S}{K} + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \qquad \text{and} \qquad \begin{aligned} d_2 &= d_1 - \sigma\sqrt{T} \\ &= \frac{\log \frac{S}{K} + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}. \end{aligned}$$

Let's recall the notation and variables in the Black-Scholes model:

Black-Scholes model

The notion and variables we're already familiar with are:

- The current date is time $t = 0$.
- C is the current European call option **premium**.
- T is the **expiry date** (and also time to expiry).
- S is the current **price of the underlying asset**.
- r is the **risk-free rate**.
- K is the **exercise** or **strike price**.

And there's two new variables that we've previously not encountered:

Black-Scholes model

First, σ is the **volatility** of the underlying asset, which we can take as the standard deviation of its continuously compounded annual returns.

- σ can be calculated from historical data as follows:

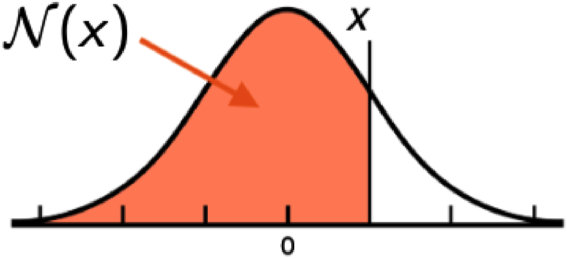
Collect a sample $\{S_0, S_1, \ldots, S_N\}$ of $N+1$ daily underlying asset prices S_i. Then calculate a sample $\{r_1, \ldots, r_N\}$ of N daily continuously compounded returns $r_i = \log \dfrac{S_i}{S_{i-1}}$. With $\bar{r} = \sum_{i=1}^{N} \dfrac{r_i}{N}$ the mean, the daily volatility is $\sigma_d = \sqrt{\sum_{i=1}^{N} \dfrac{(r_i - \bar{r})^2}{N-1}}$. The annualised volatility is then $\sigma = \sigma_d \sqrt{252}$, assuming 252 trading days in 1 year.

Black-Scholes model

Second, $\mathcal{N}(x)$ is the CDF evaluated at x of a standard normal random variable X. It gives the probability that X is less than or equal to x:

$$\boxed{\mathcal{N}(x) = \mathbb{P}(X \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} \mathrm{d}z.}$$

Graphically, it's the area under the standard normal PDF:



Black-Scholes model

Computers calculate the standard normal CDF $\mathcal{N}(x)$ for us. In R:

```
> x = 0.55
> N = pnorm(x) # pnorm() is N()
> N
[1] 0.7088403
```

In Python:

```
In [1]: from scipy.stats import norm
In [2]: x = 0.75
In [3]: N = norm.cdf(x) # SciPy's norm.cdf() is N()
In [4]: N
Out[4]: 0.7733726476231317
```

Let's calculate the Black-Scholes European call price in R and Python:

Black-Scholes model

Example

The R code

```r
S = 50
K = 50
r = 0.05
T = 1/2
sigma = 0.25
d1 = (log(S/K) + (r + 0.5*sigma^2)*T)/(sigma*sqrt(T))
d2 = d1 - sigma*sqrt(T)
C = S*pnorm(d1) - K*exp(-r*T)*pnorm(d2)
```

gives

```
> C
[1] 4.130008
```

Black-Scholes model

Example (Continued)

The Python code

```python
import numpy as np
from scipy.stats import norm
S = 50
K = 50
r = 0.05
T = 1/2
sigma = 0.25
d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T)/(sigma*np.sqrt(T))
d2 = d1 - sigma*np.sqrt(T)
C = S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)
```

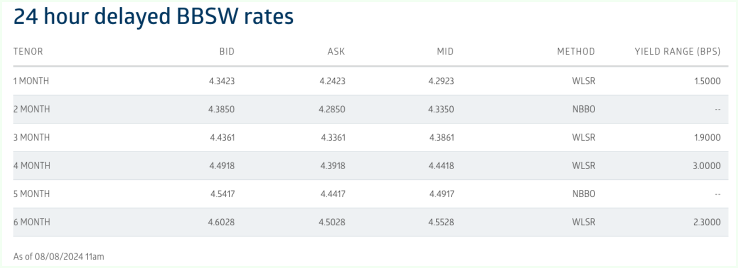
gives

```
In [1]: C
Out[1]: 4.130007599671615
```

Black-Scholes model

Let's use the yfinance Python module to calculate the historical volatility of the S&P/ASX 200 Index and price some September index option contracts, where here we'll ignore dividends (for now).

Example



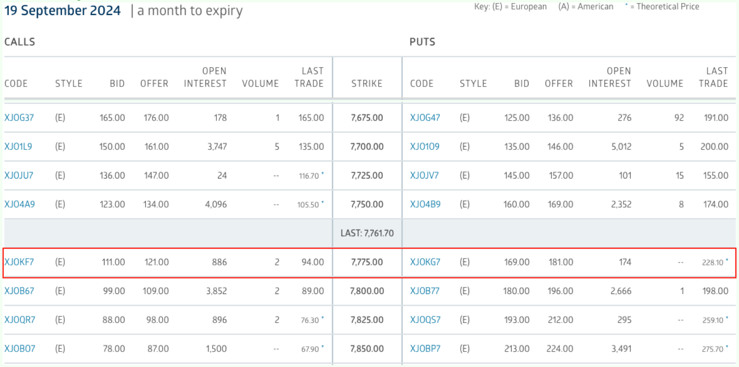
24 hour delayed BBSW rates

TENOR	BID	ASK	MID	METHOD	YIELD RANGE (BPS)
1 MONTH	4.3423	4.2423	4.2923	WLSR	1.5000
2 MONTH	4.3850	4.2850	4.3350	NBBO	--
3 MONTH	4.4361	4.3361	4.3861	WLSR	1.9000
4 MONTH	4.4918	4.3918	4.4418	WLSR	3.0000
5 MONTH	4.5417	4.4417	4.4917	NBBO	--
6 MONTH	4.6028	4.5028	4.5528	WLSR	2.3000

As of 08/08/2024 11am

Black-Scholes model

Example



19 September 2024 | a month to expiry

Key: (E) = European (A) = American * = Theoretical Price

CALLS								PUTS						
CODE	STYLE	BID	OFFER	OPEN INTEREST	VOLUME	LAST TRADE	STRIKE	CODE	STYLE	BID	OFFER	OPEN INTEREST	VOLUME	LAST TRADE
XJOG37	(E)	165.00	176.00	178	1	165.00	7,675.00	XJOG47	(E)	125.00	136.00	276	92	191.00
XJO1L9	(E)	150.00	161.00	3,747	5	135.00	7,700.00	XJO109	(E)	135.00	146.00	5,012	5	200.00
XJOJU7	(E)	136.00	147.00	24	--	116.70 *	7,725.00	XJOJV7	(E)	145.00	157.00	101	15	155.00
XJO4A9	(E)	123.00	134.00	4,096	--	105.50 *	7,750.00	XJO4B9	(E)	160.00	169.00	2,352	8	174.00
							LAST: 7,761.70							
XJOKF7	(E)	111.00	121.00	886	2	94.00	7,775.00	XJOKG7	(E)	169.00	181.00	174	--	228.10 *
XJOB67	(E)	99.00	109.00	3,852	2	89.00	7,800.00	XJOB77	(E)	180.00	196.00	2,666	1	198.00
XJOQR7	(E)	88.00	98.00	896	2	76.30 *	7,825.00	XJOQS7	(E)	193.00	212.00	295	--	259.10 *
XJOBO7	(E)	78.00	87.00	1,500	--	67.90 *	7,850.00	XJOBP7	(E)	213.00	224.00	3,491	--	275.70 *

There is 41 days until 19-Sep-2024 so we will set $T = \dfrac{41}{365}$.

Black-Scholes model

Example (Continued)

```python
import numpy as np, yfinance as yf
from scipy.stats import norm
P = yf.download("^AXJO")["Adj Close"]
ret = np.log(P).diff(1).dropna()
sigma = np.std(ret)*np.sqrt(252)
K = 7775; S = 7761.7; T = 41/365
r = (0.042923+0.04335)/2 # average of 1 & 2 month BBSW rate
d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T)/(sigma*np.sqrt(T))
d2 = d1 - sigma*np.sqrt(T)
C = S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2) # call price
P = K*np.exp(-r*T)*norm.cdf(-d2) - S*norm.cdf(-d1) # put price
```

This code gives $\sigma = 0.15375$ and European call and put prices

```
In [1]: C
Out[1]: 171.73
In [2]: P
Out[2]: 147.45
```

Black-Scholes model

In the above example we also presented put prices. The Black-Scholes European option pricing model for puts can be derived separately to calls:

$$\boxed{P = Ke^{-rT}\mathcal{N}(-d_2) - S\mathcal{N}(-d_1),}$$

where all variables and notation are as above.

- Alternatively, we can use put-call parity:

$$\boxed{P = C + e^{-rT}K - S.}$$

We check that they agree, using the original “generic” example above:

Black-Scholes model

We need only add the put equation lines to our above code.

Example

In R we add

```r
P = K*exp(-r*T)*pnorm(-d2) - S*pnorm(-d1)
P1 = C + exp(-r*T)*K - S # using put-call parity
```

to get

```
> P
[1] 2.895503
> P1
[1] 2.895503
```

Black-Scholes model

Example (Continued)

In Python we add

```python
P = K*np.exp(-r*T)*norm.cdf(-d2) - S*norm.cdf(-d1)
P1 = C + np.exp(-r*T)*K - S # using put-call parity
```

to get

```
In[1]: P
Out[1]: 2.8955
In[2]: P1
Out[2]: 2.8955
```

Incorporating dividends

We now incorporate dividends into the Black-Scholes European option pricing equations, in order to more accurately price equity options.

Incorporating dividends

Incorporating a continuously compounded dividend yield q into the basic Black-Scholes model is relatively easy. We can show that the pricing equations for European call C and put P options become

$$C = Se^{-qT}\mathcal{N}(d_1) - Ke^{-rT}\mathcal{N}(d_2),$$

$$P = Ke^{-rT}\mathcal{N}(-d_2) - Se^{-qT}\mathcal{N}(-d_1),$$

where

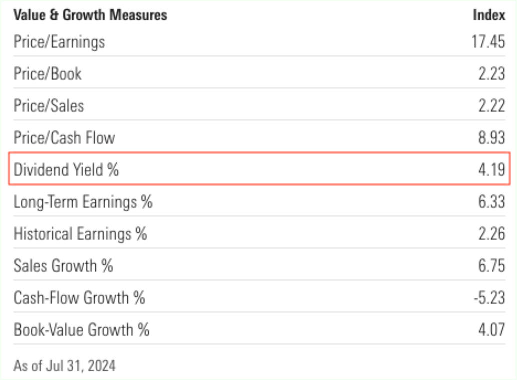
$$d_1 = \frac{\log\frac{S}{K} + (r - q + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \qquad \text{and} \qquad d_2 = d_1 - \sigma\sqrt{T}$$

$$= \frac{\log\frac{S}{K} + (r - q - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

Incorporating dividends

We continue the above example above of pricing September S&P/ASX 200 Index options, this time including a dividend yield of $q = 4.2\%$.

Example



Value & Growth Measures	Index
Price/Earnings	17.45
Price/Book	2.23
Price/Sales	2.22
Price/Cash Flow	8.93
Dividend Yield %	4.19
Long-Term Earnings %	6.33
Historical Earnings %	2.26
Sales Growth %	6.75
Cash-Flow Growth %	-5.23
Book-Value Growth %	4.07

As of Jul 31, 2024

Incorporating dividends

Example (Continued)

```python
import numpy as np, yfinance as yf
from scipy.stats import norm
XJO=yf.download("^AXJO")["Adj Close"]
ret=np.log(XJO).diff(1).dropna().rename("returns")
sigma=np.std(ret)*np.sqrt(252)
K=7775; S=7761.7; T=41/365; q=0.042 # dividend yield 4.2%
r=(0.042923+0.04335)/2 # average of 1 & 2 month BBSW rate
d1=(np.log(S/K)+(r-q+0.5*sigma**2)*T)/(sigma*np.sqrt(T))
d2=d1-sigma*np.sqrt(T)
C=S*np.exp(-q*T)*norm.cdf(d1)-K*np.exp(-r*T)*norm.cdf(d2)
P=K*np.exp(-r*T)*norm.cdf(-d2)-S*np.exp(-q*T)*norm.cdf(-d1)
```

This code gives European call and put prices

```
In [1]: C
Out[1]: 152.87
In [2]: P
Out[2]: 165.12
```

Incorporating dividends

Remark

Some reasons for inaccurate S&P/ASX 200 Index option prices:

- We're using historical volatility for σ but in practice options traders tend to use the Black-Scholes model with "rules of thumb" and their own estimates of σ.
- Our dividend yield may not be accurate: We assumed a continuously compounded yield. It should be a forecast. And should we incorporate dividend imputation/franking?
- The time stamps on our data don't align.

Currency options

We can also price currency options with the Black-Scholes model.

Currency options

When applying the Black-Scholes model to pricing currency options, we need to be mindful about the currency quoting conventions of S and K, as well as the handling of the domestic r_{d} and foreign r_{f} interest rates.

- Let's recall our currency quoting convention:

Let $S_{\mathrm{f:d}}$ and $K_{\mathrm{f:d}}$ be the spot and strike prices of 1 unit of the foreign currency (the underlying asset) in terms of the domestic currency.

The modification to the Black-Scholes model is relatively easy:

- It turns out that we view the foreign risk-free rate r_{f} as the "dividend" on the underlying asset (the foreign currency):

Currency options

$$C = S_{\mathrm{f:d}} e^{-r_{\mathrm{f}} T} \mathcal{N}(d_1) - K_{\mathrm{f:d}} e^{-r_{\mathrm{d}} T} \mathcal{N}(d_2),$$

$$P = K_{\mathrm{f:d}} e^{-r_{\mathrm{d}} T} \mathcal{N}(-d_2) - S_{\mathrm{f:d}} e^{-r_{\mathrm{f}} T} \mathcal{N}(-d_1),$$

where

$$d_1 = \frac{\log \frac{S_{\mathrm{f:d}}}{K_{\mathrm{f:d}}} + (r_{\mathrm{d}} - r_{\mathrm{f}} + \frac{1}{2}\sigma^2) T}{\sigma\sqrt{T}} \quad \text{and} \quad d_2 = d_1 - \sigma\sqrt{T}$$

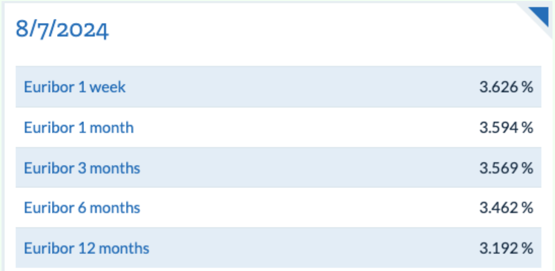
$$= \frac{\log \frac{S_{\mathrm{f:d}}}{K_{\mathrm{f:d}}} + (r_{\mathrm{d}} - r_{\mathrm{f}} - \frac{1}{2}\sigma^2) T}{\sigma\sqrt{T}}.$$

This is sometimes called the **Garman-Kohlhagen** (**GK**) equations, who suggested in their paper Foreign Currency Option Values the idea of viewing a currency option as an option on a stock paying a dividend r_{f}.

Currency options

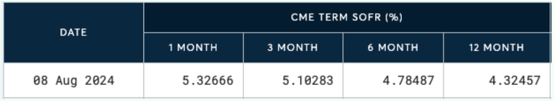
We use Python to download exchange rates and price a 3-month at-the-money EUR:USD currency option (EUR = foreign currency).

Example



8/7/2024

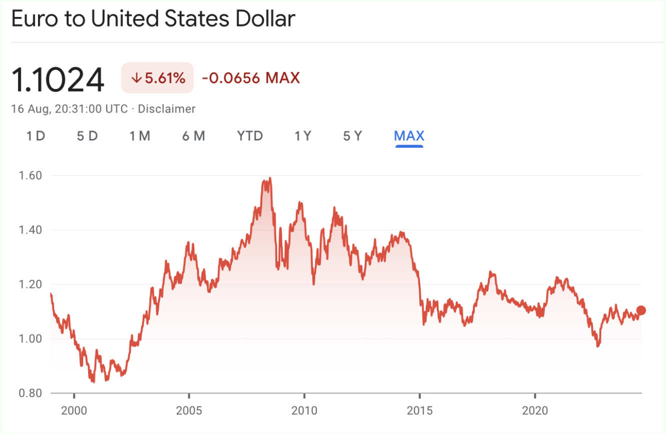
Euribor 1 week	3.626 %
Euribor 1 month	3.594 %
Euribor 3 months	3.569 %
Euribor 6 months	3.462 %
Euribor 12 months	3.192 %



DATE	CME TERM SOFR (%)			
	1 MONTH	3 MONTH	6 MONTH	12 MONTH
08 Aug 2024	5.32666	5.10283	4.78487	4.32457

Currency options

Example (Continued)


Euro to United States Dollar

1.1024 ↓5.61% -0.0656 MAX

16 Aug, 20:31:00 UTC · Disclaimer

Currency options

Example (Continued)

Let's first convert the simple Term SOFR and EURIBOR interest rates to continuous compounding. The future value of \$1 invested at a simple interest rate s is $1 + sT$. Under compound interest r, it is e^{rT}. Hence, for each of these rates, we want to find r satisfying $1 + sT = e^{rT}$. Rearranging, we get

$$r = \frac{1}{T} \log(1 + sT).$$

I calculate the continuously compounded rates to be $r_{\mathrm{f}} = 0.03553$ and $r_{\mathrm{d}} = 0.05071$.

Currency options

Example (Continued)

```
import numpy as np, yfinance as yf
from scipy.stats import norm
EURUSD = yf.download("EURUSD=X")["Adj Close"]
ret = np.log(EURUSD).diff(1).dropna()
sigma = np.std(ret)*np.sqrt(252)
T = 90/360; S = 1.1024; K = S # at-the-money
rf = (360/90)*np.log(1+0.03569*90/360) # Euro is the foreign currency
rd = (360/90)*np.log(1+0.0510283*90/360) # USD is the domestic currency
d1 = (np.log(S/K) + (rd-rf+0.5*sigma**2)*T)/(sigma*np.sqrt(T))
d2 = d1 - sigma*np.sqrt(T)
C =  S*np.exp(-rf*T)*norm.cdf(d1)  - K*np.exp(-rd*T)*norm.cdf(d2)
P = -S*np.exp(-rf*T)*norm.cdf(-d1) + K*np.exp(-rd*T)*norm.cdf(-d2)
```

This code gives $\sigma = 0.11245$ and call and put option values

```
In [1]: C
Out[1]: 0.0266
In [2]: P
Out[2]: 0.0225
```

Risk-neutral pricing and geometric Brownian motion

The ideas of risk-neutral pricing and geometric Brownian motion are central to all of quantitative finance and derivative security pricing.

- We touch on the very basics as a starting point into more advanced quantitative finance.
- We will also use both concepts for pricing options via Monte Carlo simulation in a few weeks time.

Risk-neutral pricing and geometric Brownian motion

It's important to note that the Black-Scholes model is derived under a large list of assumptions. Some of these include what might be called the "usual assumptions" in financial theory and modelling:

- Constant risk-free rate r and volatility σ over life of option.
- No restrictions on borrowing and lending.
- Borrowing and lending rates are equal.
- No transaction costs like brokerage, bid-ask spreads, taxes, etc.
- Assets are infinitely divisible (can buy "1.0346" units).
- No restrictions on short selling.
- Trading is continuous (at infinitesimally small time intervals).
- There are no arbitrage opportunities in markets.

Risk-neutral pricing and geometric Brownian motion

We're not so much interested in them as we are in the assumption on the stochastic or random process followed by the underlying asset.

- This assumption basically characterises the Black-Scholes framework and tells us the correct interpretation of the volatility parameter σ.

In the risk-neutral pricing approach, we can prove that the underlying asset follows **geometric Brownian motion** (**GBM**)

$$S_t = Se^{(r-\frac{1}{2}\sigma^2)t+\sigma\sqrt{t}Z} \qquad \text{for } 0 \leq t \leq T,$$

where Z is a standard normal random variable. This implies that the underlying asset's returns are normally distributed:

Risk-neutral pricing and geometric Brownian motion

Rearranging the above, the asset's continuously compounded returns

$$r_t = \log \frac{S_t}{S} = \left(r - \frac{1}{2}\sigma^2\right)t + \sigma\sqrt{t}Z$$

are normally distributed with $\mathbb{E}[r_t] = \left(r - \frac{1}{2}\sigma^2\right)t$ and $\mathbb{V}\text{ar}[r_t] = \sigma^2 t$.

- This is the proper interpretation of σ, namely that it's the "diffusion" parameter in geometric Brownian motion.

Remark

Rearranging: $\log S_t = \log S + \left(r - \frac{1}{2}\sigma^2\right)t + \sigma\sqrt{t}Z$. So $\log S_t$ is normally distributed with mean $\mathbb{E}[S_t] = \log S_t + \left(r - \frac{1}{2}\sigma^2\right)t$ and variance $\mathbb{V}\text{ar}[r_t] = \sigma^2 t$. This is what is meant by the stock price S_t being **log-normally distributed**.

Risk-neutral pricing and geometric Brownian motion

In the risk-neutral approach, the underlying asset follows geometric Brownian motion as presented above. We can simulate it as follows:

- Discretise the interval $[0, T]$ into $M + 1$ equally-spaced dates $\{t_0, t_1, \ldots, t_M\}$ with $t_0 = 0$, $t_M = T$ and spacing $\mathrm{d}t = \frac{T}{M}$.
- Let S_j be the underlying price at date t_j, with $S_0 = S$.
- Iteratively simulate geometric Brownian motion over each subinterval $[t_{j-1}, t_j]$ to create a complete path by setting

$$\boxed{S_j = S_{j-1} e^{(r - \frac{1}{2}\sigma^2)\mathrm{d}t + \sigma\sqrt{\mathrm{d}t} Z_j} \qquad \text{for } j = 1, \ldots, M.}$$

Each Z_j is an independent standard normal random variable.

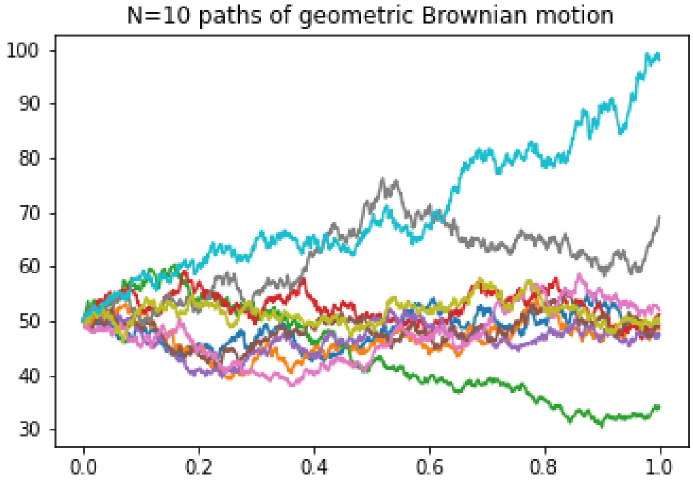
Risk-neutral pricing and geometric Brownian motion

Simulating GBM by the above is used in the Monte Carlo simulation pricing of options, and we do this later in the course. Here is some Python code that simulates and plots geometric Brownian motion paths:

Example (Continued)

```python
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
S0=50; r=0.05; T=1; sigma=0.25
M=1000; dt=T/M; dates=np.linspace(0,T,M+1) # 1000 time steps
N=10; S=np.zeros([N,M+1]); S[:,0]=S0 # 10 paths, each starting at S=50
for i in range(N):
    for j in range(1,M+1):
        Z=norm.rvs() # simulate outcomes of standard normal random variable
        S[i,j]=S[i,j-1]*np.exp((r-0.5*sigma**2)*dt+sigma*np.sqrt(dt)*Z) # GBM
    plt.plot(dates,S[i,:])
plt.title("N=10 paths of geometric Brownian motion") # plot
```

Risk-neutral pricing and geometric Brownian motion



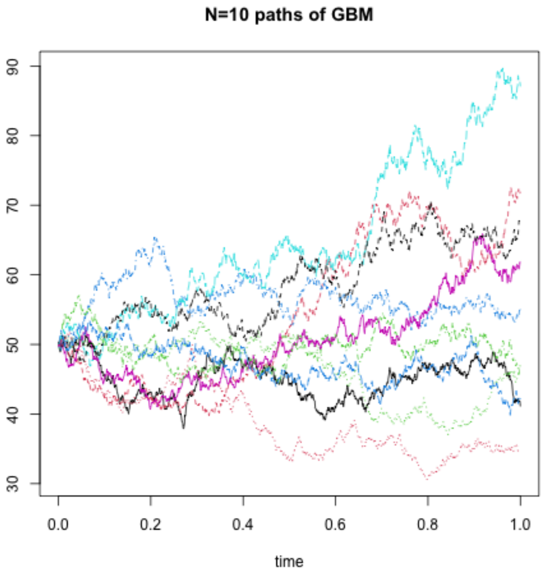
Risk-neutral pricing and geometric Brownian motion

And here's some R code the simulates and plots GBM:

Example (Continued)

```r
# we don't need to load any libraries
S0=50; r=0.05; T=1; sigma=0.25
M=1000; dt=T/M; dates=seq(0,1,dt) # 1000 time steps
N=10; S=matrix(0,N,M+1); S[,1]=S0 # 10 paths, each starting at S=50
for(i in 1:N)
{
  for(j in 1:M)
  {
    Z=rnorm(1) # simulate outcomes of standard normal random variable
    S[i,j+1]=S[i,j]*exp((r-0.5*sigma^2)*dt+sigma*sqrt(dt)*Z) # GBM
  }
}
matplot(dates,t(S),type="l",main="N=10 paths of GBM", xlab="time",ylab="") # plot
```

Risk-neutral pricing and geometric Brownian motion



Risk-neutral pricing and geometric Brownian motion

But what is meant by the "risk-neutral" pricing approach? First note:

Law of finance: The value of an asset is the present value of its expected futures cashflows or payoff.

We can use a lot of complex maths to show that the value of European options in the risk-neutral approach is given by the law of finance:

$$C = e^{-rT}\mathbb{E}\big[\max\{0, S_T - K\}\big] \quad \text{and} \quad P = e^{-rT}\mathbb{E}\big[\max\{0, K - S_T\}\big]$$

where S_T is log-normally distributed as specified above.

Risk-neutral pricing and geometric Brownian motion

So in the risk-neutral pricing approach the value of a European option is

- the present value of its expected future payoff
- but discounted at the risk-free rate r.

Remark

So in the risk-neutral pricing approach, investors don't add a risk premium to the discount rate, since it's just the risk-free rate r.

- Investors are "neutral" to risk.

We can also rederive the futures/forward pricing formulas again here in the risk-neutral approach:

Risk-neutral pricing and geometric Brownian motion

Remark

The payoff of a long futures contract is $S_T - K$, so the value of a long futures contract in the risk-neutral approach is

$$V = e^{-rT}\mathbb{E}[S_T - K] = e^{-rT}\mathbb{E}[S_T] - e^{-rT}K.$$

But K is set so these contracts have 0 value: $V = 0$. In the risk-neutral approach, the value of *every* asset is the present value of its expected future payoff, including the underlying asset, so $S = e^{-rT}\mathbb{E}[S_T]$. So using this and rearranging the above yields

$$K = e^{rT}S.$$

Variables affecting option prices and the Greeks

We now want to investigate and quantify how each of the input variables K, S, r, T, σ and q impact option premiums.

Variables affecting option prices and the Greeks

The Black-Scholes model

$$C = S\mathcal{N}(d_1) - Ke^{-rT}\mathcal{N}(d_2) \quad \text{and} \quad P = Ke^{-rT}\mathcal{N}(-d_2) - S\mathcal{N}(-d_1)$$

(ignoring dividends q for now), where

$$d_1 = \frac{\log \frac{S}{K} + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \qquad \text{and} \qquad d_2 = d_1 - \sigma\sqrt{T},$$

depends on the variables S, K, r, T and σ.

- We're interested in the relation between them and option prices.

We know that call (put) options with a higher strike price K have lower (higher) values, and we're not so much interested any further in K.

Variables affecting option prices and the Greeks

We're interested in the sensitivity of option prices to the other variables S, r, T and σ, and we give these sensitivities special Greek names:

- **Delta** Δ is the sensitivity of the option price to S.
- **Gamma** Γ is another measure of the relation between prices and S.
- **Rho** ρ is the sensitivity of the option price to rates r.
- **Vega** ν is the sensitivity of the option price to volatility σ.
- **Theta** θ is the sensitivity of the option price to time T.

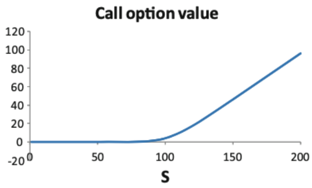
We cover each of these one-by-one.

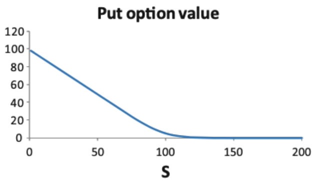
Remark

We will assume no dividends q since the equations are neater, until right at the end where we mention the impact of dividends.

Delta Δ and gamma Γ

We know that as S increases, calls premiums rise and put premiums fall.





We quantify this mathematically with the **delta** Δ, given by

$$\Delta_C = \mathcal{N}(d_1) \qquad \text{and} \qquad \Delta_P = \mathcal{N}(d_1) - 1.$$

Importantly, note the following, which confirms the first line of this slide:

$$0 < \Delta_C < 1 \qquad \text{and} \qquad -1 < \Delta_P < 0.$$

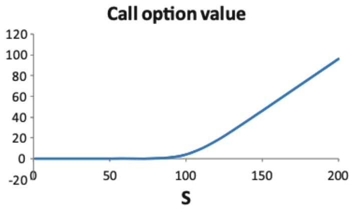
Delta Δ and gamma Γ

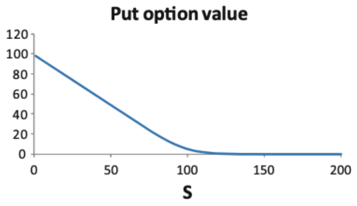
Remark

Δ is the partial derivative of the premium with respect to S.

But think of Δ as:

- The change in the premium due to a change in S.
- In fact, Δ is the <u>slope</u> of the above payoff diagrams:





Delta Δ and gamma Γ

From the remark above, we interpret Δ as the change in the premium due to a change in S, giving us the approximations

$$\mathrm{d}C \approx \Delta_C \mathrm{d}S \qquad \text{and} \qquad \mathrm{d}P \approx \Delta_P \mathrm{d}S,$$

where $\mathrm{d}C$ and $\mathrm{d}P$ are a change in the premium and $\mathrm{d}S$ a change in S.

- We calculate the "new" premiums due to a change in S as

$$C_{\text{new}} \approx C + \mathrm{d}C \qquad \text{and} \qquad P_{\text{new}} \approx P + \mathrm{d}P.$$

Remark

These approximations are used later in delta hedging.

Delta Δ and gamma Γ

Example

In the original example we had $S = K = 50$, $r = 5\%$, $T = 1/2$ and $\sigma = 25\%$. This yielded $C = 4.13$ and $P = 2.8955$. We get

$$\Delta_C = 0.591 \qquad \text{and} \qquad \Delta_P = -0.41.$$

Let's now suppose $S = 52$ so $\mathrm{d}S = 2$. We calculate that

$$\mathrm{d}C \approx \Delta_C \mathrm{d}S = 1.182.$$

The new call price is $C_{\text{new}} \approx C + \mathrm{d}C = 5.312$.

- If $S = 52$ in the Black-Scholes equation, we get $C = 5.397$.

Delta Δ and gamma Γ

From above, the approximation $\mathrm{d}C \approx \Delta_C \mathrm{d}S$ is not perfect, but:

- We can make it more accurate by also using the **gamma** Γ given by

$$\Gamma = \frac{f(d_1)}{S\sigma\sqrt{T}} \qquad \text{(same for calls and puts)},$$

with $f(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}$ the PDF of a standard normal random variable.

Remark

Γ is the 2nd partial derivative of the premium with respect to S.

Delta Δ and gamma Γ

We make the approximations of $\mathrm{d}C$ and $\mathrm{d}P$ more accurate by setting

$$\mathrm{d}C \approx \Delta_C \mathrm{d}S + \frac{1}{2}\Gamma \mathrm{d}S^2 \qquad \text{and} \qquad \mathrm{d}P \approx \Delta_P \mathrm{d}S + \frac{1}{2}\Gamma \mathrm{d}S^2.$$

Example (Continued)

Continuing with the above example, we calculate that

$$\Gamma = 0.04396.$$

The change in the call price is $\mathrm{d}C \approx \Delta_C \mathrm{d}S + \frac{1}{2}\Gamma \mathrm{d}S^2 = 1.2697$ so the new call price is $C_{\text{new}} \approx C + \mathrm{d}C = 5.3997$.

Delta Δ and gamma Γ

Example (Continued)

Some Python code to calculate this example is:

```python
import numpy as np
from scipy.stats import norm
S = 50; K = 50; r = 0.05; T = 1/2; sigma = 0.25
d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T)/(sigma*np.sqrt(T))
d2 = d1 - sigma*np.sqrt(T)
C = S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)
deltaC = norm.cdf(d1); deltaP = deltaC - 1
gamma = norm.pdf(d1)/(S*sigma*np.sqrt(T)) # norm.pdf() is f()
dS = 2
dC = deltaC*dS + 0.5*gamma*dS**2
dP = deltaP*dS + 0.5*gamma*dS**2
Cnew = C + dC; Pnew = P + dP
```

Rho ρ

Rho ρ is the change in the premium from a change in r, and is given by

$$\rho_C = KTe^{-rT}\mathcal{N}(d_2) \qquad \text{and} \qquad \rho_P = KTe^{-rT}\left[\mathcal{N}(d_2) - 1\right].$$

Importantly, note that

$$\boxed{0 < \rho_C \qquad \text{and} \qquad \rho_P < 0.}$$

<u>As r increases, call premiums increase but put premiums decrease</u>.

- An intuitive explanation for this is the present value $e^{-rT}K$ of the strike that the holder pays in a call, or receives in a put, decreases.

Rho ρ

Remark

ρ is the partial derivative of the premium with respect to r.

Example (Continued)

Following on with the same example, we calculate that

$$\rho_C = 12.707 \qquad \text{and} \qquad \rho_P = -11.676.$$

The Python code would be

```
rhoC = K*T*np.exp(-r*T)*norm.cdf(d2)
rhoP = K*T*np.exp(-r*T)*(norm.cdf(d2) - 1)
```

Vega ν

Vega ν is the change in the premium from a change in σ, and is given by

$$\nu = Sf(d_1)\sqrt{T} \qquad \text{(same for calls and puts)},$$

with $f(x)$ the PDF of a standard normal random variable. Note that

$$0 < \nu.$$

As σ increases, option premiums increase.

Remark

ν is the partial derivative of the premium with respect to σ.

Vega ν

Example

Following on from the previous example, we calculate that

$$\nu = 13.737.$$

The Python code would be

```
vega = S*norm.pdf(d1)*np.sqrt(T)
```

Theta θ

Theta θ is a bit ambiguous. It gives the *negative* of the change in the premium from a change in T. And the equations are more complex:

$$\theta_C = -\frac{Sf(d_1)\sigma}{2\sqrt{T}} - rKe^{-rT}\mathcal{N}(d_2),$$

$$\theta_P = \theta_C + rKe^{-rT},$$

with $f(x)$ the PDF of a standard normal random variable.

Remark

θ is the *negative* of the partial derivative of the premium with respect to T, telling us the impact of approaching expiry.

Theta θ

However, note that

$$\theta_C < 0 \qquad \text{but we may have} \quad \theta_P \leq 0 \quad \text{or} \quad 0 \leq \theta_P.$$

On a non-dividend paying asset, call premiums fall as expiry nears.

- Impact of time on puts is ambiguous: From $\theta_P = \theta_C + rKe^{-rT}$ deep in-the-money put (large K) premiums may *increase* as expiry nears.
  - This relates to something said last week: A deep in-the-money put is already close to its maximum payoff of K, so not much more payoff can be realised at expiry, but there is still a chance of an unfavourable movement in the asset price. But as we approach expiry, there is less chance of an unfavourable movement.

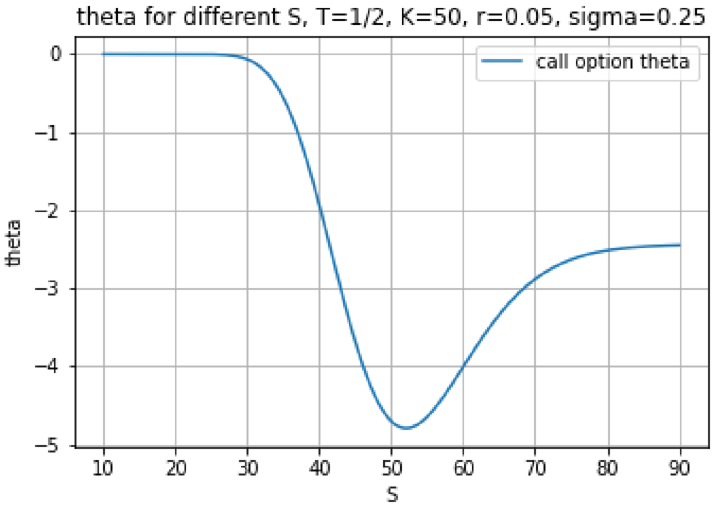
Theta θ

However, there is some rules of thumb relating to time:

- θ is almost always negative:
    - So premiums usually fall as expiry approaches.
    - This is known as **time decay**.
    - Time decay works for option writers and against option holders.
- θ is most negative for options close to at-the-money.
- θ in fact gets more negative for options close to at-the-money as expiry approaches: Time decay "speeds up" as expiry approaches.

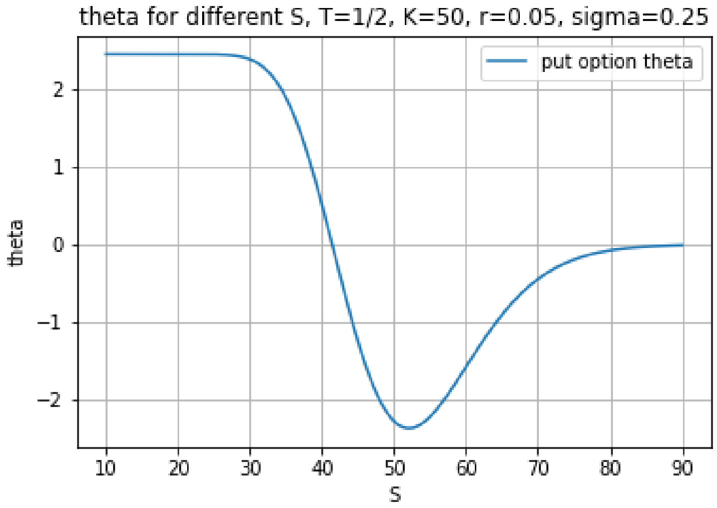
These last two points are illustrated in the following figures:

Theta θ



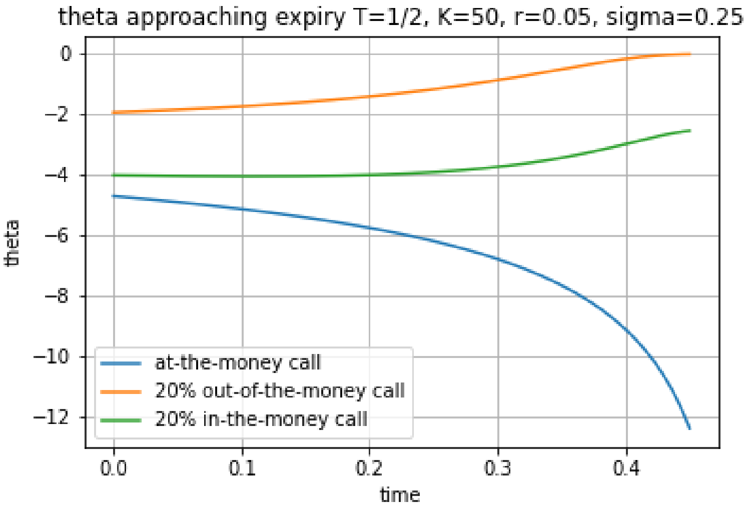
theta for different S, T=1/2, K=50, r=0.05, sigma=0.25
call option theta
theta
S
0
−1
−2
−3
−4
−5
10
20
30
40
50
60
70
80
90

Theta θ

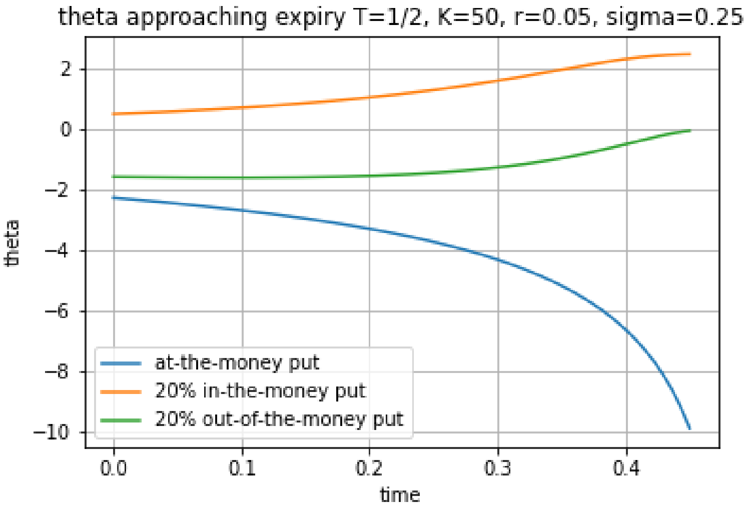


theta for different S, T=1/2, K=50, r=0.05, sigma=0.25
put option theta
theta
S
2
1
0
−1
−2
10
20
30
40
50
60
70
80
90

Theta θ


theta approaching expiry T=1/2, K=50, r=0.05, sigma=0.25
theta
time
0
−2
−4
−6
−8
−10
−12
0.0
0.1
0.2
0.3
0.4
at-the-money call
20% out-of-the-money call
20% in-the-money call

Theta θ



theta approaching expiry T=1/2, K=50, r=0.05, sigma=0.25
theta
time
0.0
0.1
0.2
0.3
0.4
2
0
−2
−4
−6
−8
−10
at-the-money put
20% in-the-money put
20% out-of-the-money put

Theta θ

Example

Continuing with the same example, we calculate that

$$\theta_C = -4.705 \qquad \text{and} \qquad \theta_P = -2.267,$$

noting that $S = K = 50$, so our options are at-the-money.

- The Python code would be

```
thetaC = -S*norm.pdf(d1)*sigma/(2*np.sqrt(T)) - r*K*np.exp(-r*T)*norm.cdf(d2)
thetaP = thetaC + r*K*np.exp(-r*T)
```

Incorporating dividends

On a dividend paying asset, all of the above interpretations remain unchanged except for θ. From the textbook, the equations become:

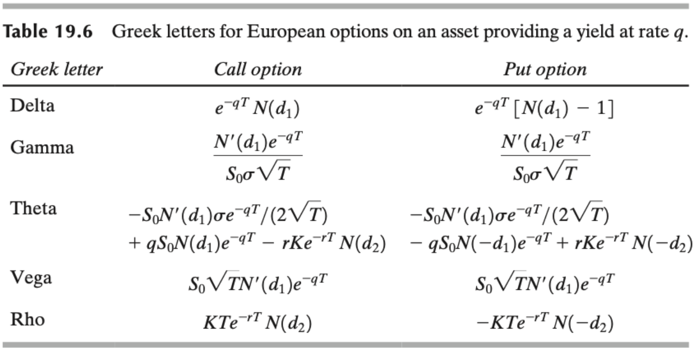


Table 19.6 Greek letters for European options on an asset providing a yield at rate q.

Greek letter	*Call option*	*Put option*
Delta	$e^{-qT} N(d_1)$	$e^{-qT} [N(d_1) - 1]$
Gamma	$\dfrac{N'(d_1)e^{-qT}}{S_0\sigma\sqrt{T}}$	$\dfrac{N'(d_1)e^{-qT}}{S_0\sigma\sqrt{T}}$
Theta	$-S_0N'(d_1)\sigma e^{-qT}/(2\sqrt{T})$ $+ qS_0N(d_1)e^{-qT} - rKe^{-rT} N(d_2)$	$-S_0N'(d_1)\sigma e^{-qT}/(2\sqrt{T})$ $- qS_0N(-d_1)e^{-qT} + rKe^{-rT} N(-d_2)$
Vega	$S_0\sqrt{T}N'(d_1)e^{-qT}$	$S_0\sqrt{T}N'(d_1)e^{-qT}$
Rho	$KTe^{-rT} N(d_2)$	$-KTe^{-rT} N(-d_2)$

Note that Hull uses the notation $N'(x)$ for the PDF of a standard normal random variable, whereas I've been using $f(x)$.

Incorporating dividends

Remark

From the above table, the theta θ_C of a European call on a dividend paying stock may also be positive or negative so the impact of time on call premiums here is also ambiguous, but:

- Don't forget the rule of thumb that θ is almost always negative, so time decay is almost always working for the option writer and against the option taker/holder.
- And time decay tends to "speed up" as expiry approaches, particularly for options close to at-the-money.

Incorporating dividends

Finally, note that we calculate that the change in premiums from a change in the dividend yield q are given by

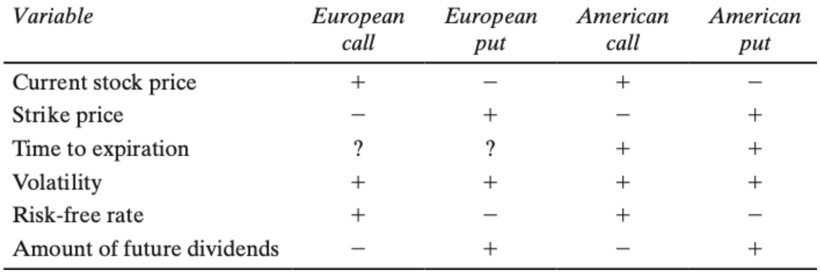
$$\frac{\partial C}{\partial q} = -TSe^{-qT}\mathcal{N}(d_1) \qquad \text{and} \qquad \frac{\partial P}{\partial q} = -TSe^{-qT}\left[\mathcal{N}(d_1) - 1\right].$$

The important interpretation is

$$\boxed{\frac{\partial C}{\partial q} < 0 \qquad \text{and} \qquad 0 < \frac{\partial P}{\partial q}.}$$

<u>As q increases, call premiums decrease but put premiums increase</u>.

Summary of variables affecting option prices



Variable	*European call*	*European put*	*American call*	*American put*
Current stock price	+	−	+	−
Strike price	−	+	−	+
Time to expiration	?	?	+	+
Volatility	+	+	+	+
Risk-free rate	+	−	+	−
Amount of future dividends	−	+	−	+

Summary