

---

# **ImpVol Documentation**

***Release 0.1***

**Stanislav Khrapov**

**May 01, 2018**



---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Examples</b>	<b>3</b>
<b>3</b>	<b>Functions</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



The original Black-Scholes formula is given by

$$\begin{aligned}BS(S, K, \sigma, r, T) &= S\Phi(d_1) - e^{-rT}K\Phi(d_2), \\d_1 &= \frac{\log(S/K) + rT}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}, \\d_2 &= d_1 - \sigma\sqrt{T}.\end{aligned}$$

After normalization by the current asset price  $S$  it can be written as

$$\begin{aligned}\tilde{BS}(X, \sigma, T) &= \Phi(d_1) - e^X\Phi(d_2), \\d_1 &= -\frac{X}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}, \\d_2 &= d_1 - \sigma\sqrt{T},\end{aligned}$$

where  $X = \log(K/F)$  is log-forward moneyness, and forward price is given by  $F = Se^{rT}$ .



## CHAPTER 2

---

### Examples

---

```
>>> from impvol import imp_vol, lfmoneyness
>>> strike = [1, .95]
>>> premium = [.024, .057]
>>> price = 1
>>> riskfree = .02
>>> maturity = 30/365
>>> call = True
>>> moneyness = lfmoneyness(price, strike, riskfree, maturity)
>>> vol = imp_vol(moneyness, maturity, premium, call)
>>> print(vol)
[ 0.20277309  0.20093061]
>>> vol = impvol_bisection(moneyness, maturity, premium, call)
>>> print(vol)
[ 0.20270996  0.20095215]
```





`impvol.impvol.imp_vol (moneyness, maturity, premium, call)`

Compute implied volatility given vector of option premium.

### Parameters

**moneyness** [array\_like] Log-forward moneyness

**maturity** [array\_like] Fraction of the year

**premium** [array\_like] Option premium normalized by current asset price

**call** [bool array\_like] Call/put flag. True for call, False for put

### Returns

**array\_like** Implied volatilities. Shape of the array is according to broadcasting rules.

### Notes

This code relies on SciPy root method. Although vectorized, it is still very slow. Bisection method in this `impvol` library is substantially faster.

`impvol.impvol.impvol_bisection (moneyness, maturity, premium, call, tol=1e-05, fcount=1000)`

Function to find BS Implied Vol using Bisection Method.

### Parameters

**moneyness** [array\_like] Log-forward moneyness

**maturity** [array\_like] Fraction of the year

**premium** [array\_like] Option premium normalized by current asset price

**call** [array\_like bool] Call/put flag. True for call, False for put

### Returns

**array\_like** Implied volatilities Shape of the array is according to broadcasting rules.

`impvol.impvol.impvol_table(data)`

Implied volatility for structured data.

### Parameters

**data** [pandas DataFrame, record array, or dictionary of arrays] Mandatory labels: moneyness, maturity, premium, call

### Returns

**array** Implied volatilities

### Notes

‘premium’ should be normalized by the current asset price.

`impvol.impvol.lfmoneyness(price, strike, riskfree, maturity)`

Compute log-forward moneyness.

### Parameters

**price** [array\_like] Underlying prices

**strike** [array\_like] Option strikes

**riskfree** [array\_like] Annualized risk-free rate

**maturity** [array\_like] Time horizons, in shares of the calendar year

### Returns

**array\_like** Log-forward moneyness

`impvol.impvol.blackscholes_norm(moneyness, maturity, vol, call)`

Standardized Black-Scholes Function.

### Parameters

**moneyness** [array\_like] Log-forward moneyness

**maturity** [array\_like] Fraction of the year, i.e. = 30/365

**vol** [array\_like] Annualized volatility (sqrt of variance), i.e. = .15

**call** [bool array\_like] Call/put flag. True for call, False for put

### Returns

**array\_like** Option premium standardized by current asset price

**i**

`impvol.impvol`, 1



## B

`blackscholes_norm()` (in module `impvol.impvol`), 6

## I

`imp_vol()` (in module `impvol.impvol`), 5

`impvol.impvol` (module), 1

`impvol_bisection()` (in module `impvol.impvol`), 5

`impvol_table()` (in module `impvol.impvol`), 5

## L

`lfmoneyness()` (in module `impvol.impvol`), 6