tcb/box tcb/boxSect
tcb/box/begin0 tcb/box/begindefault
tag=tcb/box tcb/box/begindefault
tcb/box/end0 tcb/box/enddefault
tcb/box/enddefault
tcb/box/title tcb/box/titleCaption
tcb/box/title0 tcb/box/titledefault para/semantictcb/box/title tcb/box/titledefault
tcb/box/draw2 tcb/box/drawdefault 2
tcb/box/drawdefault
tcb/box/init0 tcb/box/initdefault minipage/beforenoopminipage/afternoop
tcb/box/initdefault
tcb/box/upper0 tcb/box/upperdefault minipage/beforetag/dfltminipage/aftertag/dflt
tcb/box/upperdefault
tcb/box/lower0 tcb/box/lowerdefault minipage/beforetag/dfltminipage/aftertag/dflt
tcb/box/lowerdefault
tcb/drawing/init0tcb/drawing/initdefaulttcb/drawing/initdefault

Python Implementation

```
def black_scholes_call(S, K, T, r, sigma):
    # Calculate European call option price
    d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) \
         / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    return S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)
```

Delta Calculation

```
def delta(S, K, T, r, sigma):
    d1 = calculate_d1(...)
    return norm.cdf(d1)
```

- Delta $= N(d_1)$
- Range: $[0, 1]$ for calls
- Measures price sensitivity

All Greeks

```
# Delta
delta = norm.cdf(d1)

# Gamma
gamma = norm.pdf(d1) / \
    (S * sigma * sqrt(T))
```

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

# Theta
theta = -S *
    norm.pdf(d1) * \
     sigma / (2 * sqrt(T))
```

Example Usage

```
# Example: Price a call option
S = 100       # Stock price
K = 105       # Strike
T = 0.5       # Time (years)
r = 0.05      # Risk-free rate
sigma = 0.2   # Volatility

price = black_scholes_call(S, K, T, r, sigma)
print(f"Call price: ${price:.2f}")
# Output: Call price: $5.57
```

Inline Code

You can also use inline code like `import numpy as np` within text.

The function `black_scholes_call S K T r sigma)` takes five parameters:

- `S` – Current stock price
- `K` – Strike price
- `T` – Time to expiration (in years)
- `r` – Risk-free interest rate
- `sigma` – Volatility