Derivatives and Fixed Income — Lecture Notes

Program: PGE M1 — Quantitative Finance Track
Skema Business School
Academic Year 2025/26 — Fall 2025
Lecturer: Alexandre Landi

1 From (Stock + Put) to (Call + Bond): Put–Call Parity

We now turn to one of the most important identities in option pricing — the **Put–Call Parity**.

Step 1: Protective Put — The Insurance Analogy

To understand the intuition behind **put–call parity**, let us begin with a concrete numerical example.

Example (illustrative data):

- Underlying stock: Generic equity (modeled after Apple Inc.)
- Current price: $S_0 = 200$ USD
- Strike price: $K = 200$ USD
- Maturity: $T = 3$ months (0.25 years)
- Risk-free rate: $r = 5\%$
- Volatility: $\sigma = 20\%$
- Market put premium: $P_0 = 10$ USD

Suppose a trader:

- Buys one share of stock at $200, and
- Buys one European put option with strike $200 and the same maturity for $10.

This combination is called a **protective put**:

$$\text{Protective Put} = \text{Long Stock} + \text{Long Put}.$$

—

Economic Intuition: Insurance for Your Portfolio

The protective put acts as a form of price insurance:

- If the stock price rises above \$200, the trader participates fully in the upside, just as with the stock alone.
- If the stock price falls below \$200, the put option offsets losses by guaranteeing the right to sell at \$200.

In essence, the trader has paid a \$10 "insurance premium" to ensure that the portfolio cannot lose more than \$10 per share at maturity.

$$\text{Maximum loss} = P_0 = \$10, \qquad \text{Potential gain} = \text{unlimited}.$$

—

Numerical Illustration

At maturity:

- If $S_T = 150$, the stock loses \$50, but the put gains \$50. Total P&L $= -50+50-10 = -\$10$.
- If $S_T = 200$, both the stock and put break even on price, but the premium still costs \$10 $\rightarrow$ P&L $= -\$10$.
- If $S_T = 250$, the stock gains \$50, the put expires worthless, net P&L $= +50 - 10 = +\$40$.

Thus, the trader's losses are limited to the put premium, but gains remain unlimited.

$$\boxed{\text{Minimum P\&L} = -\$10, \quad \text{Break-even} = S_T = 210, \quad \text{Maximum P\&L} = \infty.}$$

—

Visual Interpretation

The figure below shows the P&L at maturity for:

- the stock alone (blue dashed line),
- the put alone (orange dashed line), and
- the combined protective put (black solid line).

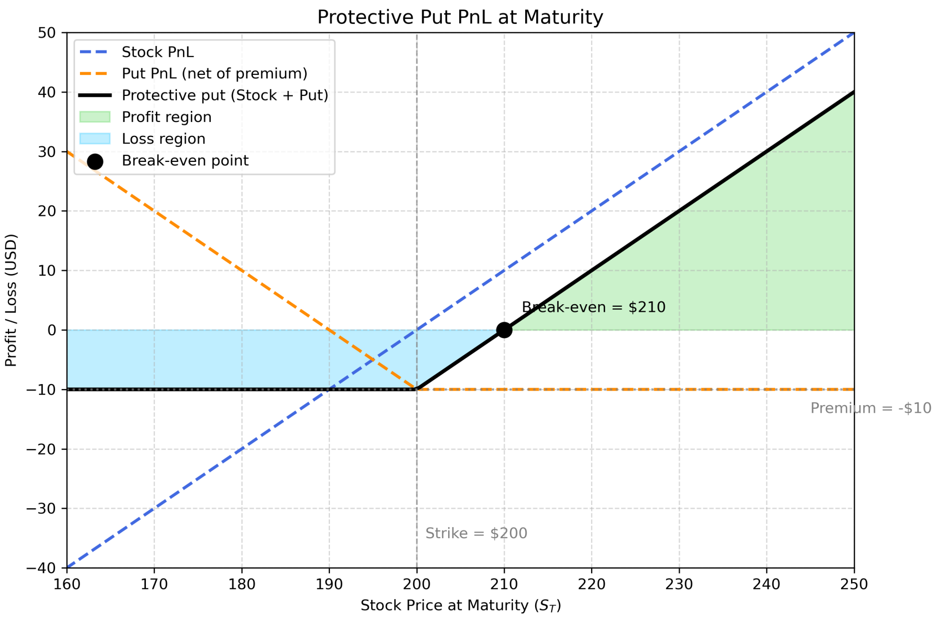

Figure 1: Protective Put P&L at Maturity ($S_0 = 200$, $K = 200$, Premium=$10).

The shaded blue area represents potential losses, while the green area represents profits. The large black dot marks the **break-even point** at $210, where the trader recovers both the stock cost and the option premium.

$$\text{Break-even: } S_T = S_0 + P_0 = 200 + 10 = 210.$$

This payoff profile mirrors the logic of a call option with a guaranteed floor — a key insight that directly leads to the concept of **put–call parity**.

This combination is called a **protective put**:

$$\text{Protective Put} = \text{Long Stock} + \text{Long Put}.$$

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters (simplified Apple-style example)
S0 = 200        # Current stock price
K = 200         # Strike price
put_premium = 10  # Cost of the put option

# Range of stock prices at maturity
S_T = np.linspace(150, 260, 200)

# PnL (Profit and Loss) calculations
stock_pnl = S_T - S0
put_pnl = np.maximum(K - S_T, 0) - put_premium
protective_put_pnl = stock_pnl + put_pnl

# Plot setup
plt.figure(figsize=(9,6))
plt.plot(S_T, stock_pnl, label='Stock PnL', color='royalblue',
         linewidth=2, linestyle='--')
```

```
plt.plot(S_T, put_pnl, label='Put PnL (net of premium)',
         color='darkorange', linewidth=2, linestyle='--')
plt.plot(S_T, protective_put_pnl, label='Protective put (Stock + Put)',
         color='black', linewidth=2.5)

# Shade profit/loss regions
plt.fill_between(S_T, protective_put_pnl, 0,
                 where=(protective_put_pnl > 0),
                 color='limegreen', alpha=0.25, label='Profit region')
plt.fill_between(S_T, protective_put_pnl, 0,
                 where=(protective_put_pnl < 0),
                 color='deepskyblue', alpha=0.25, label='Loss region')

# Break-even point
breakeven = S0 + put_premium  # For a protective put

# Big black dot at the breakeven point
plt.scatter(breakeven, 0, color='black', s=100, zorder=5, label='Break-even point')
plt.text(breakeven + 2, 3, f'Break-even = ${breakeven:.0f}', color='black', fontsize=10)

# Annotations
plt.axvline(K, color='gray', linestyle='--', alpha=0.7, linewidth=1)
plt.text(K + 1, -35, f'Strike = ${K}', color='gray', fontsize=10)
plt.axhline(-put_premium, color='gray', linestyle=':', alpha=0.7)
plt.text(245, -put_premium - 4, f'Premium = -${put_premium}', color='gray', fontsize=10)

# Formatting
plt.title("Protective Put PnL at Maturity", fontsize=13)
plt.xlabel("Stock Price at Maturity ($S_T$)")
plt.ylabel("Profit / Loss (USD)")
plt.legend(loc='upper left')
plt.grid(True, linestyle='--', alpha=0.5)
plt.xlim(160, 250)
plt.ylim(-40, 50)
plt.tight_layout()

# Save or display
plt.savefig("protective_put_pnl.png", dpi=300)
```

Listing 1: Python code to plot the Protective Put P&L at maturity

—

Step 2: Replicating the Same P&L — Call + Bond Combination

We can replicate the same **P&L profile** as the protective put using a different mix of instruments. The key relationship is:

$$\text{Stock} + \text{Put} \iff \text{Call} + \text{Risk-Free Bond}.$$

—

Economic Intuition

- Below the strike price K, the **put** in the protective-put portfolio guarantees a minimum value of \$200. In the replication, this role is played by a **zero-coupon bond** that pays \$200 at maturity.
- Above K, both portfolios move one-for-one with the stock price because the trader either holds the stock directly (in the protective put) or owns a call that becomes active above the strike (in the replication).

Thus, the two portfolios are *identical at maturity* and, by the principle of no arbitrage, must have the same value today:

$$\boxed{\text{Stock} + \text{Put} = \text{Bond} + \text{Call.}}$$

The bond here is a **zero-coupon bond** maturing in 3 months with a face value of \$200 (the strike). If the continuously compounded risk-free rate is 5%, its current price is:

$$B_0 = K\,e^{-rT} = 200\,e^{-0.05\times 0.25} \approx 197.52.$$

—

Numerical Illustration

Replication portfolio:

- Buy one European call with strike \$200.
- Buy one risk-free zero-coupon bond worth \$197.52 today, paying \$200 at maturity.

At maturity:

$$\text{Payoff} = \max(S_T - 200, 0) + 200.$$

Examples:

- If $S_T = 150$, the call expires worthless and the bond pays \$200 → P&L = –\$10 (cost of call).
- If $S_T = 200$, the call is at the money; trader still receives \$200 from the bond → P&L = –\$10.
- If $S_T = 250$, the call gains \$50, the bond pays \$200 → P&L = +\$40.

These results are identical to the **protective put** outcomes from Step 1.

—

Visual Comparison

The figure below shows that the **Call + Bond** portfolio (red) perfectly matches the **Protective Put** (black). The horizontal portion of the curve (the floor) comes from the bond, and the upward-sloping portion (the upside) from the call.

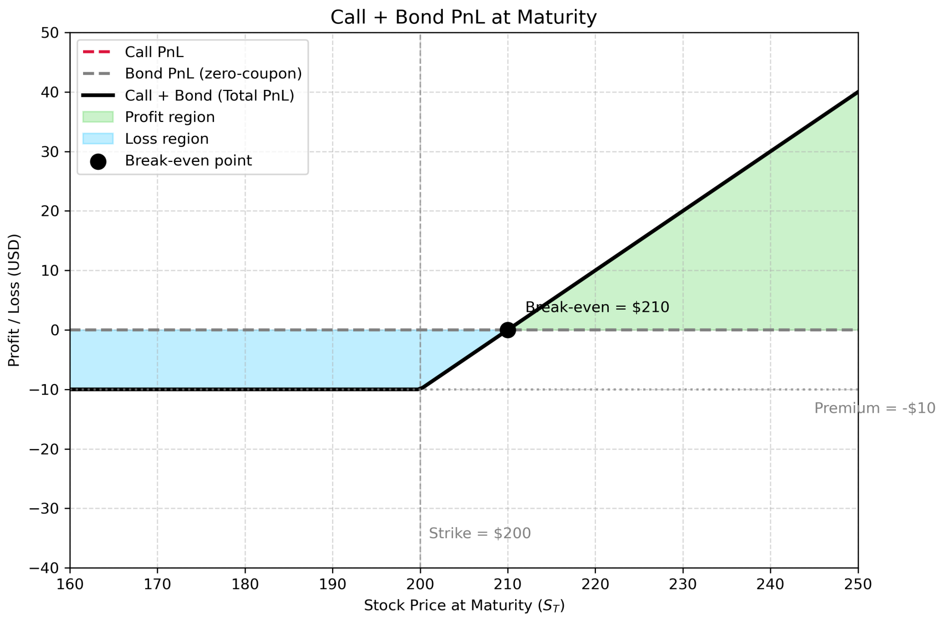

Figure 2: Replication of the Protective Put using a Call + Bond combination ($S_0 = 200$, $K = 200$, $r = 5\%$, $T = 0.25$).

.

```python
import numpy as np
import matplotlib.pyplot as plt

# Parameters (simplified Apple-style example)
S0 = 200          # Current stock price
K = 200           # Strike price
r = 0.05          # Risk-free rate
T = 0.25          # Time to maturity (in years)
call_premium = 10   # Cost of the call option

# Present value of the zero-coupon bond that pays K at maturity
bond_price = K
bond_payout = K                       # Maturity value

# Range of stock prices at maturity
S_T = np.linspace(150, 260, 200)

# PnL (Profit and Loss) calculations
call_pnl = np.maximum(S_T - K, 0) - call_premium
bond_pnl = (bond_payout - bond_price) * np.ones_like(S_T)
call_bond_pnl = call_pnl + bond_pnl

# Plot setup
plt.figure(figsize=(9,6))
plt.plot(S_T, call_pnl, label='Call PnL', color='crimson',
         linewidth=2, linestyle='--')
plt.plot(S_T, bond_pnl, label='Bond PnL (zero-coupon)',
         color='gray', linewidth=2, linestyle='--')
plt.plot(S_T, call_bond_pnl, label='Call + Bond (Total PnL)',
         color='black', linewidth=2.5)

# Shade profit/loss regions
```

```python
plt.fill_between(S_T, call_bond_pnl, 0,
                 where=(call_bond_pnl > 0),
                 color='limegreen', alpha=0.25, label='Profit region')
plt.fill_between(S_T, call_bond_pnl, 0,
                 where=(call_bond_pnl < 0),
                 color='deepskyblue', alpha=0.25, label='Loss region')

# Break-even point
breakeven = K + call_premium - (bond_payout - bond_price)

# Big black dot at the breakeven point
plt.scatter(breakeven, 0, color='black', s=100, zorder=5, label='Break-
    even point')
plt.text(breakeven + 2, 3, f'Break-even = ${breakeven:.0f}', color='
    black', fontsize=10)

# Annotations
plt.axvline(K, color='gray', linestyle='--', alpha=0.7, linewidth=1)
plt.text(K + 1, -35, f'Strike = ${K}', color='gray', fontsize=10)
plt.axhline(-call_premium, color='gray', linestyle=':', alpha=0.7)
plt.text(245, -call_premium - 4, f'Premium = -${call_premium}', color='
    gray', fontsize=10)

# Formatting
plt.title("Call + Bond PnL at Maturity", fontsize=13)
plt.xlabel("Stock Price at Maturity ($S_T$)")
plt.ylabel("Profit / Loss (USD)")
plt.legend(loc='upper left')
plt.grid(True, linestyle='--', alpha=0.5)
plt.xlim(160, 250)
plt.ylim(-40, 50)
plt.tight_layout()

# Save or display
plt.savefig("call_bond_pnl.png", dpi=300)
plt.show()
```

Listing 2: Python code to plot the Call + Bond P&L at Maturity

—

2 Interlude: The Zero-Coupon Bond

A **zero-coupon bond** is the simplest type of fixed-income instrument:

- It pays a single amount (the *face value*) at maturity.
- It does not pay any intermediate coupons.

—

Example (Matching the Option Pricing Setup)

In our earlier example, the strike price is $K = 200$, the continuously compounded risk-free rate is $r = 5\%$, and the time to maturity is $T = 0.25$ years (3 months).

The present value of a bond that will pay \$200 at maturity is:

$$B_0 = Ke^{-rT} = 200\, e^{-0.05\times 0.25} \approx 197.53.$$

Thus, an investor can buy a zero-coupon bond today for approximately \$197.53 to receive \$200 in three months.

In general, we can say that before maturity, the present value of a bond that pays K at time T is therefore:

$$B = Ke^{-rT}.$$

—

Interpretation

This bond acts as a **risk-free floor**: no matter what happens to the stock or option prices, it will pay \$200 at maturity.

In the context of **put–call parity**, this bond is the component that ensures the guaranteed payoff in the **Call + Bond** portfolio:

$$\boxed{\text{Stock} + \text{Put} \;=\; \text{Bond} + \text{Call.}}$$

—

3 Step 3: Formal Statement of Put–Call Parity

At maturity (T):

$$S_T + P_T = K + C_T.$$

Before maturity (time $t = 0$):

$$\boxed{S_0 + P_0 = C_0 + Ke^{-rT}.}$$

This equality must hold to avoid arbitrage. It links the prices of European options with the same strike and maturity.

Alternative Forms

By rearranging:

$$\begin{aligned} C_0 &= S_0 + P_0 - Ke^{-rT}, \\ P_0 &= C_0 + Ke^{-rT} - S_0, \\ C_0 - P_0 &= S_0 - Ke^{-rT}. \end{aligned}$$

Each form is algebraically equivalent and expresses the same pricing identity.

—

4 Step 4: Numerical Example and Arbitrage

Let us now check whether the put–call parity relationship holds numerically, using the same data as before.

$$\begin{aligned} S_0 &= 200, \\ K &= 200, \\ r &= 5\%, \quad T = 0.25, \quad e^{-rT} = e^{-0.05\times 0.25} \approx 0.9876, \\ C_0 &= 10, \\ P_0 &= 10. \end{aligned}$$

—

Step 4.1: Compute Both Sides of Put–Call Parity

$$\text{Value of Stock + Put} = 200 + 10 = 210,$$

$$\begin{aligned} \text{Value of Call + Bond} &= 10 + 200\,e^{-0.05\times 0.25} \\ &= 10 + 200(0.9876) \\ &= 10 + 197.52 \\ &= 207.52. \end{aligned}$$

The two sides are not equal:

$$210 > 207.52.$$

Thus, the **Stock + Put** portfolio is overpriced relative to the **Call + Bond** portfolio.

—

Step 4.2: Arbitrage Strategy

- **Sell (short)** the overpriced portfolio: short the stock and **write** (sell) the put.
- **Buy** the cheaper portfolio: go long one call and one zero-coupon bond paying \$200 at maturity.

$$\text{Cash flow at initiation: } +210 - 207.52 = +2.48.$$

This \$2.48 profit is earned immediately and is completely risk-free.

—

Step 4.3: Scenario Check at Maturity

Case 1: Stock price falls to \$150.

- The call expires worthless.
- The put (which we sold) is exercised — we must buy the stock at \$200.
- The bond we own pays \$200 at maturity, perfectly covering this obligation.
- We keep the initial \$2.48 arbitrage profit.

Case 2: Stock price rises to \$250.

- The put expires worthless.
- The call we own is exercised — worth \$50.
- The bond pays \$200, giving a total of \$250.
- We use this amount to buy back the shorted stock.
- Again, no loss — the initial \$2.48 profit is retained.

—

Conclusion

In both cases, the outcome at maturity is identical and risk-free. The initial cash inflow of \$2.48 is a pure arbitrage profit, demonstrating that when put–call parity does not hold, traders can construct a **zero-risk arbitrage** by selling the overpriced portfolio and buying the cheaper one.

—

5 Interpretation

- The parity ensures consistent pricing between puts and calls.
- If it fails, arbitrageurs act immediately to restore equality.
- In practice, such mispricings are rare and short-lived because algorithms exploit them almost instantly.
- The relation holds exactly for **European options**. For **American options**, early exercise breaks this symmetry.

—

6 Identities

$$\boxed{\begin{aligned} S_0 + P_0 &= C_0 + Ke^{-rT}, \\ C_0 - P_0 &= S_0 - Ke^{-rT}, \\ C_0 &= S_0 + P_0 - Ke^{-rT}, \\ P_0 &= C_0 + Ke^{-rT} - S_0. \end{aligned}}$$

These expressions form the backbone of modern option pricing. They guarantee that call and put prices remain internally consistent with the price of the underlying and the risk-free rate.

—

7 Summary

- A protective put behaves like a call plus a bond.
- The zero-coupon bond connects option payoffs across time.

- The put–call parity ensures arbitrage-free relationships:

$$S_0 + P_0 = C_0 + Ke^{-rT}.$$

- If violated, an arbitrage opportunity arises — allowing risk-free profit.