Md. Raihan Tapader & Efat Khan

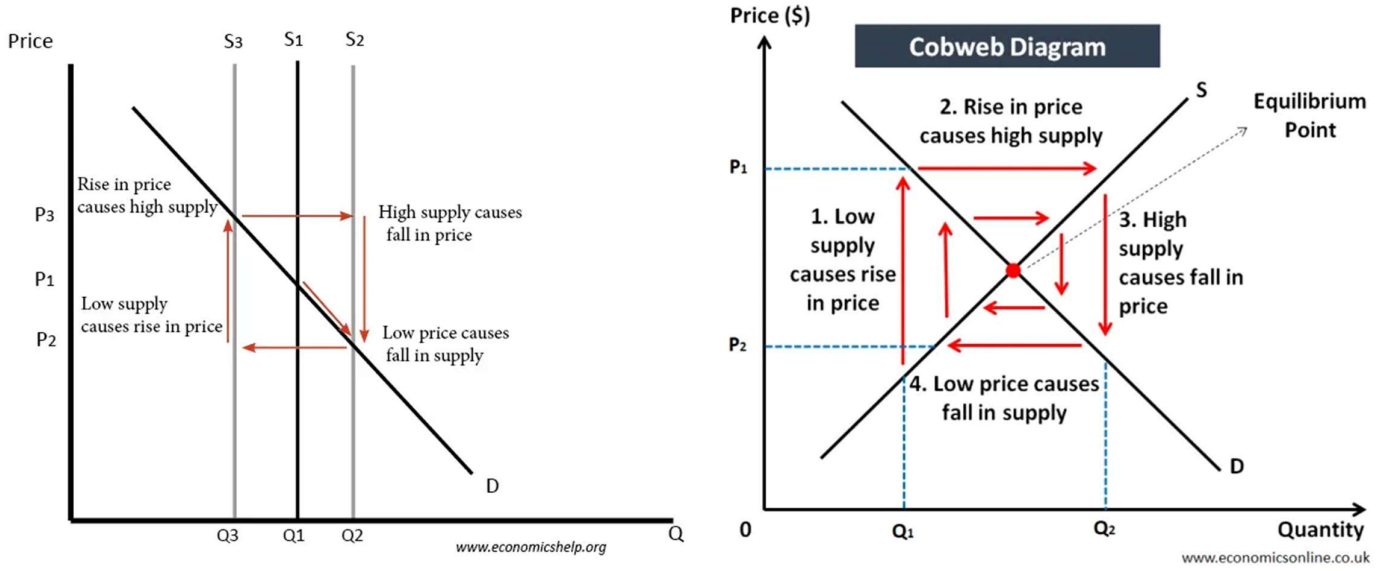
ID:204006, 204071

Cobweb Model

Introduction: ***Nicholas Kaldor*** introduced the ***"cobweb theorem"*** in 1934, building on earlier analyses by Henry Schultz and Umberto Ricci, originally published in German.

The cobweb theory describes how price fluctuations can lead to changes in supply, creating a cyclical pattern of rising and falling prices.

In its simplest form, the cobweb model examines an agricultural market, where supply is influenced by unpredictable factors, such as weather conditions.



Cobweb Model Graph

The graph illustrates the **Cobweb Model**, showing how price and supply fluctuate in cycles:

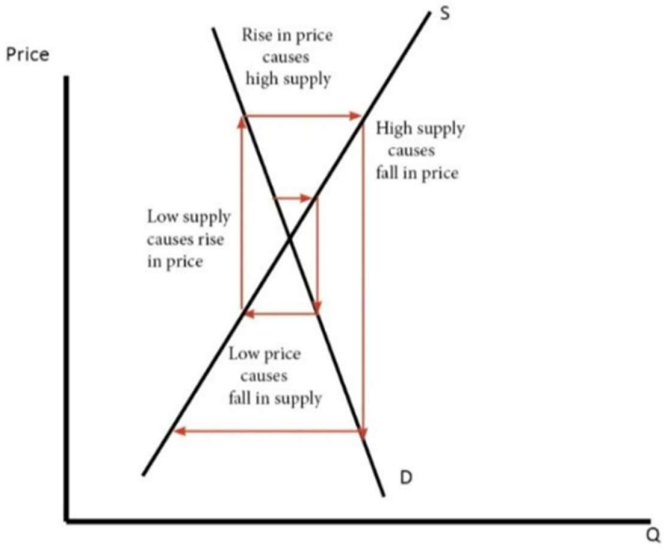
1. **Starting Point (Q1, P1)**: At ***equilibrium***, price and supply intersect at **P1** and **Q1**.

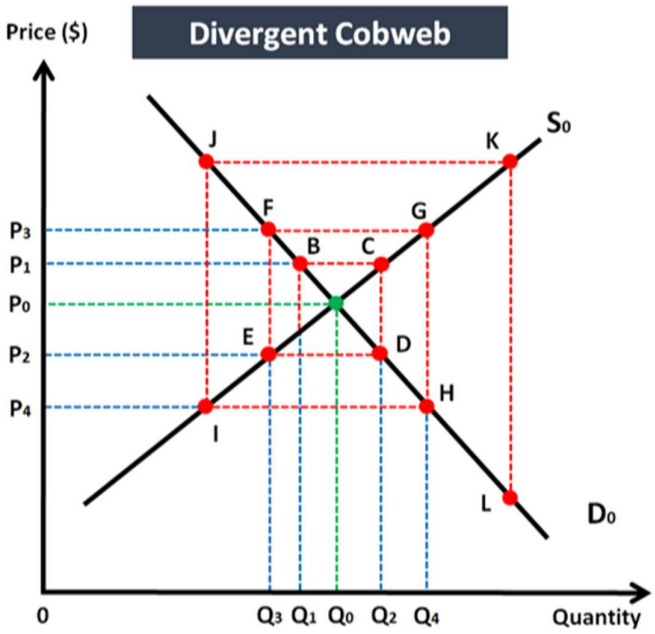
Cycle Explanation:

1. **Starting Point (Q1, P1)**:
   - At equilibrium, price and supply intersect at **P1** and **Q1**.
2. **Low Supply → Price Rises** (S3 → P3):
   - If supply drops to **Q3** due to unexpected conditions (e.g., bad weather), the price increases to **P3**.
3. **High Price → High Supply** (S3 → S2):
   - Encouraged by higher prices, farmers increase supply for the next year to **Q2**, shifting the supply curve right to **S2**.
4. **High Supply → Price Falls** (P2):
   - The increase in supply leads to a surplus, causing prices to drop to **P2**.

5. **Low Price → Supply Falls** (S2 → S3):
    - The low price discourages production, causing supply to reduce again to **Q3**, repeating the cycle.

<u>Price divergence:</u> If the slope of **the supply curve is less than the demand curve**, then the price changes could become magnified and the market more unstable.

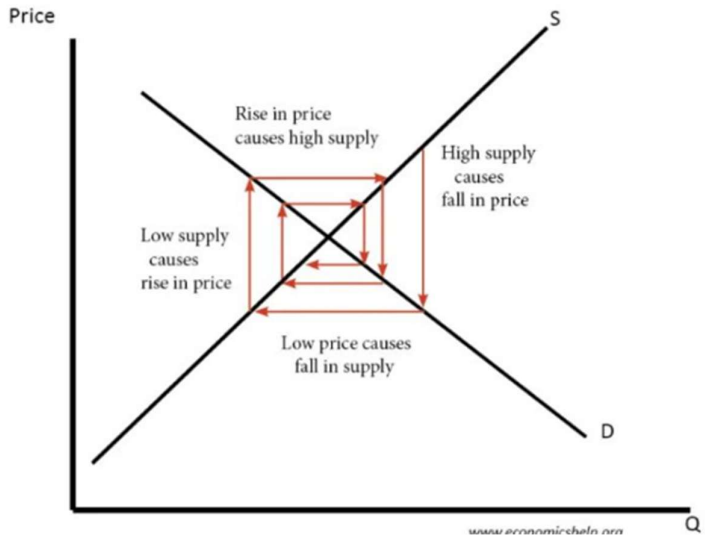


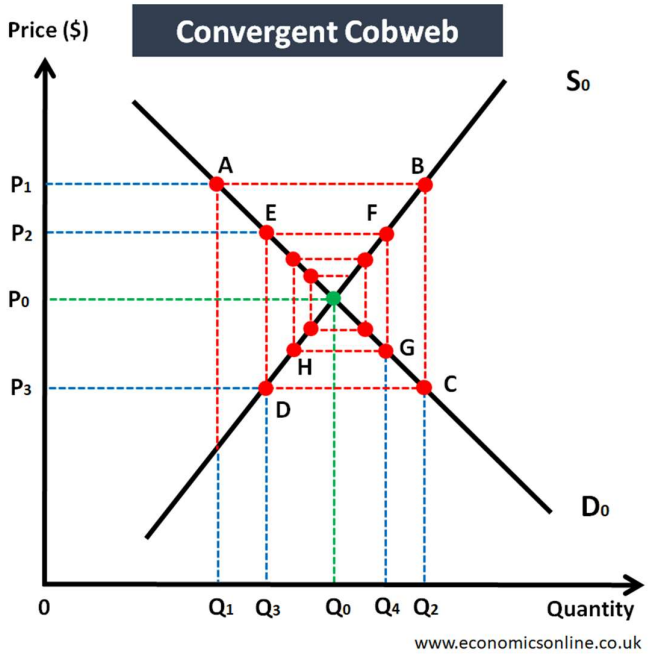


Note: 1. If the **supply curve** tilts **more horizontally** compared to the demand curve, it is flatter (**smaller slope**).

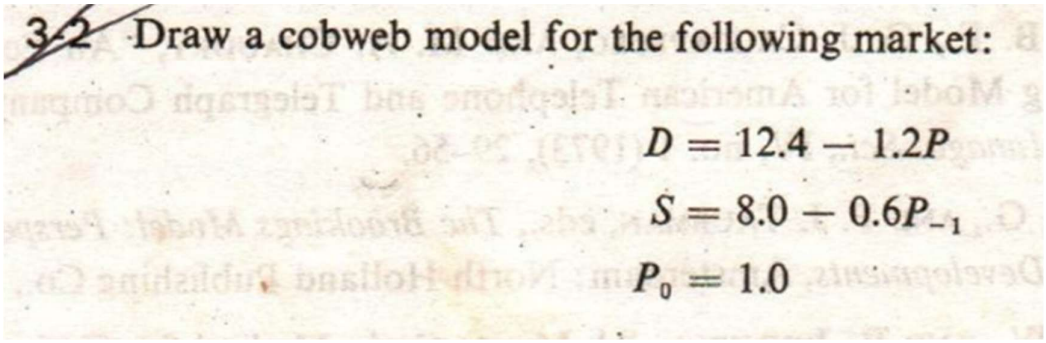
2. If the **demand curve** is **more vertical**, it is steeper.

<u>Price convergence:</u> At the **equilibrium point**, if the **demand curve is more elastic** (flatter) than the supply curve, price changes have smaller effects on demand compared to supply.





Title: Solve the cobweb model problem



3.2 Draw a cobweb model for the following market:

$$D = 12.4 - 1.2P$$

$$S = 8.0 - 0.6P_{-1}$$

$$P_0 = 1.0$$

Solution:

At Equlibrium point,

For Particular solution , let **Pt = Pt-1 = P**

D = S so,

12.4-1.2Pt= 8.0-0.6P$t-1$

12.4-8.0 = 1.2P - 0.6P

P = (12.4-8.0) / (1.2 - 0.6)

so **P* or P = 7.33**

Putting the value of P in any equation of S or D ,

Now, Q* = 12.4 - 1.2 * 7.33

Q* = 3.60

So equilibrium point is (7.33 , 3.60)

When t=1 means (First Year)

Given **P0** = 1.0

Q1 = Qs1 = 8.0 - 0.6 P(t-1) = 8.0 - 0.6 P(0) = 8.0 - 0.6 * 1.0 = 7.4

Qd1 = Q1 = 12.4 - 1.2 Pt

7.4 = 12.4 - 1.2 P1

P1 = 4.167

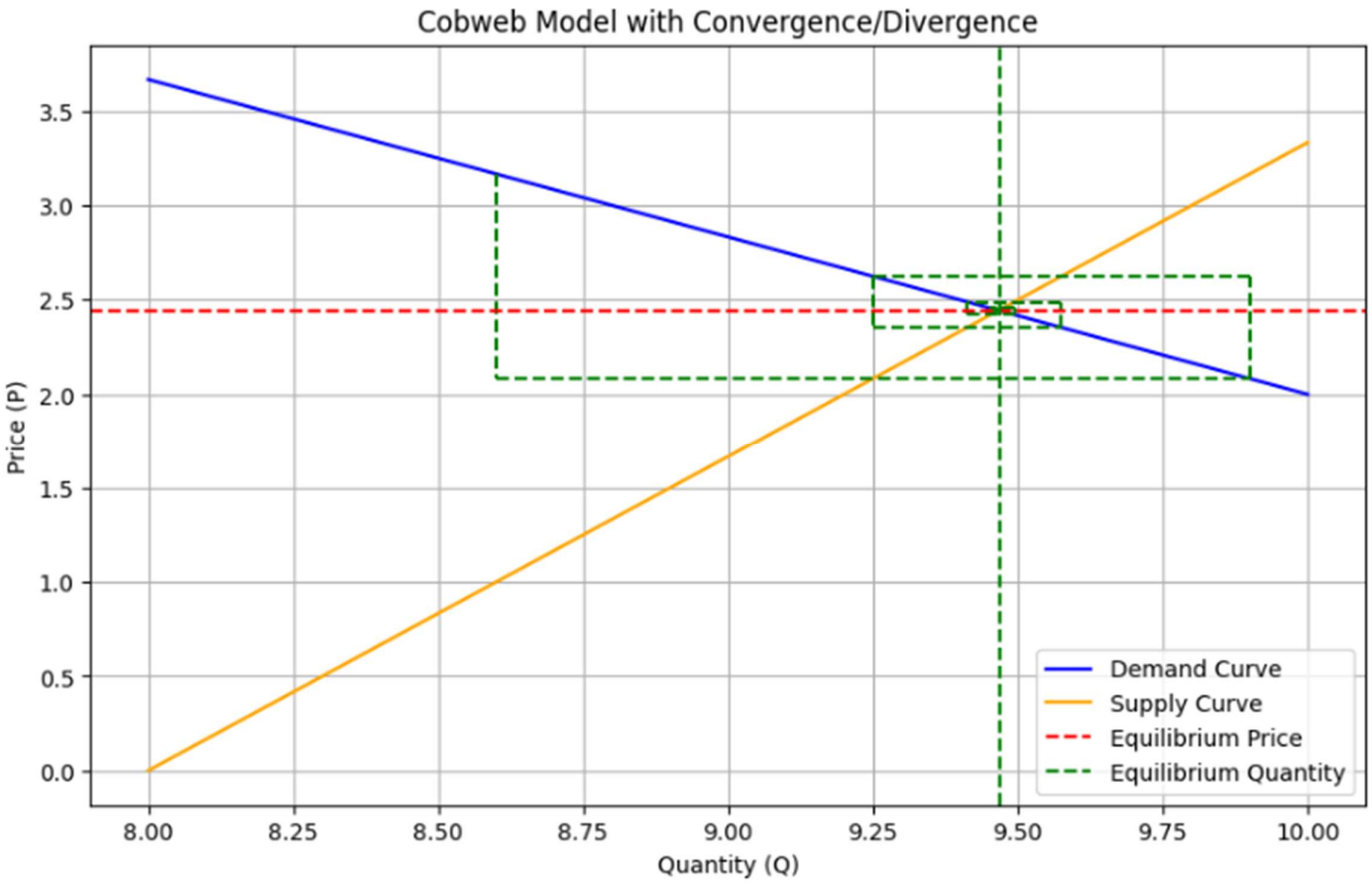
.....

Code:

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

# Parameters for demand and supply functions
demand_intercept = 12.4
demand_slope = 1.2
supply_intercept = 8.0
supply_slope = 0.6

# Initial price
P_prev = 1.0  # Initial price (P0)
iterations = 10  # Number of iterations

# Lists to store values for plotting
prices = []
quantities = []

# Iterative calculation for the cobweb path
for i in range(iterations):
    # Calculate quantity supplied based on previous price
    Q_supply = supply_intercept + supply_slope * P_prev

    # Calculate price for the current demand = supply
    P_current = (demand_intercept - Q_supply) / demand_slope

    # Store values for plotting
    prices.append(P_current)
    quantities.append(Q_supply)

    # Update previous price for the next iteration
    P_prev = P_current

# Print the price and quantity at each iteration
print("Iteration\tQuantity (Q)\tPrice (P)")
for i in range(len(prices)):
    print(f"{i+1}\t\t{quantities[i]:.4f}\t\t{prices[i]:.4f}")

# Demand curve function
```

```
def demand_curve(Q):
    return (demand_intercept - Q) / demand_slope

# Supply curve function
def supply_curve(Q):
    return (Q - supply_intercept) / supply_slope

# Generate quantities for the demand and supply curves
Q_values = np.linspace(8, 10, 100)
P_demand = demand_curve(Q_values)
P_supply = supply_curve(Q_values)

# Function to plot the cobweb
def plot_cobweb(quantities, prices, demand_curve, supply_curve):
    plt.figure(figsize=(10, 6))

    # Plot demand and supply curves
    Q_values = np.linspace(8, 10, 100)
    plt.plot(Q_values, demand_curve(Q_values), label="Demand Curve", color="blue")
    plt.plot(Q_values, supply_curve(Q_values), label="Supply Curve", color="orange")

    # Plot cobweb path
    for i in range(len(quantities) - 1):
        # Vertical line (from price to supply curve)
        plt.plot([quantities[i], quantities[i]], [prices[i], prices[i + 1]],
color="green", linestyle="--")

        # Horizontal line (from supply curve to demand curve)
        plt.plot([quantities[i], quantities[i + 1]], [prices[i + 1], prices[i + 1]],
color="green", linestyle="--")

    # Add equilibrium lines
    plt.axhline(y=(12.4 - 8.0) / (1.2 + 0.6), color="red", linestyle="--",
label="Equilibrium Price")
    plt.axvline(x=9.47, color="green", linestyle="--", label="Equilibrium Quantity")

    # Labeling axes and adding title
    plt.xlabel("Quantity (Q)")
    plt.ylabel("Price (P)")
    plt.title("Cobweb Model with Convergence/Divergence")
    plt.legend()
    plt.grid()
    plt.show()

# Plot the demand and supply curves with the cobweb path
plot_cobweb(quantities, prices, demand_curve, supply_curve)
```

Graph:


Cobweb Model with Convergence/Divergence
Price (P)
Quantity (Q)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
8.00
8.25
8.50
8.75
9.00
9.25
9.50
9.75
10.00
Demand Curve
Supply Curve
Equilibrium Price
Equilibrium Quantity