Київський національний університет

імені Тараса Шевченка

Лабораторна робота №1

з предмету “ Екологічні і економічні процеси та їх моделювання ”

Виконав

Студент 3 курсу

Групи ТТП-32

Факультету комп’ютерних наук

та кібернетики

Ходаков Максим

Київ

07.03.2023

**1. Побудова графіків функцій попиту та пропозиції.**

Функції попиту та пропозиції були визначені за допомогою степеневих функцій. Отримані параметри для функції попиту a приблизно 302.18, b приблизно 1.2 а для функції пропозиції: с приблизно 48.14, d приблизно 1.10. Функція попиту: f = 302.18 \* P ^ (-1.25) Функція пропозиції: g = 48.14 \* P ^ (1.10)

A screen shot of a graph

Description automatically generated

**2. Побудувати знайдені функції в осях (Q, P). Знайти точку ринкової рівноваги, нанести її на графік. Дослідити стан рівноваги на стабільність.**

Точка ринкової рівноваги була знайдена з ціною приблизно P=2.18 та кількістю Q=113.81. Це означає, що за цієї ціни кількість товару, яку бажають купити споживачі, дорівнює кількості товару, яку виробники бажають продати.

A screen shot of a graph

Description automatically generated

**3. Визначити дугову еластичність попиту та пропозиції на всьому діапазоні спостережень.**

У точці рівноваги похідна кривої попиту дорівнює приблизно −65.24, а похідна кривої пропозиції дорівнює приблизно 57.49. Абсолютна величина похідної кривої попиту більша, ніж абсолютна величина похідної кривої пропозиції у точці рівноваги.

A black screen with white text

Description automatically generated

**4. З’ясувати та графічно відобразити, як змінюватимуться параметри ринкової рівноваги після введення субсидії. Вивести на графік нові ціни споживача та виробника.**

З введенням субсидії у розмірі 0.5, ринкова рівновага змістилася. Нова ціна споживача становить приблизно P=2.18, тоді як ціна виробника, з урахуванням субсидії, зменшилася до приблизно P=1.68. Нова кількість у точці рівноваги збільшилася до приблизно Q=85.43. Субсидія призвела до зниження виробничої ціни та збільшення кількості товару, що продається.

**A screen shot of a graph

Description automatically generated**

**5. Код програми**

import pandas as *pd*import numpy as *np*from scipy.optimize import curve\_fit, fsolve  
import matplotlib  
  
matplotlib.use('TkAgg')  
import matplotlib.pyplot as *plt*# Функції для апроксимації кривих попиту і пропозиції  
def demand\_func(*p*, *a*, *b*):  
 return *a* \* np.power(*p*, -*b*)  
  
  
def supply\_func(*p*, *c*, *d*):  
 return *c* \* np.power(*p*, *d*)  
  
  
# Функція для знаходження точки ринкової рівноваги  
def equilibrium(*p*, *a\_demand*, *b\_demand*, *c\_supply*, *d\_supply*):  
 *Qd* = demand\_func(*p*, *a\_demand*, *b\_demand*)  
 *Qs* = supply\_func(*p*, *c\_supply*, *d\_supply*)  
 return *Qs* - *Qd*# Щоб дослідити стабільність рівноваги, ми повинні дивитися на крутизну (похідні) кривих попиту та пропозиції  
# в точці рівноваги. Рівновага стабільна, якщо крутизна кривої пропозиції більша за крутизну кривої попиту  
# в точці рівноваги (тобто, |dQs/dP| > |dQd/dP|).  
  
# Визначимо функції для похідних  
def demand\_derivative(*p*, *a*, *b*):  
 return -*a* \* *b* \* np.power(*p*, -(*b* + 1))  
  
  
def supply\_derivative(*p*, *c*, *d*):  
 return *c* \* *d* \* np.power(*p*, (*d* - 1))  
  
  
# Дугова еластичність обчислюється як відсоткове зміщення кількості, ділене на відсоткове зміщення ціни  
# для двох точок на кривій. Формула для дугової еластичності:  
# E = [(Q2 - Q1) / ((Q2 + Q1) / 2)] / [(P2 - P1) / ((P2 + P1) / 2)]  
  
def arc\_elasticity(*Q1*, *Q2*, *P1*, *P2*):  
 *quantity\_percent\_change* = (*Q2* - *Q1*) / ((*Q2* + *Q1*) / 2)  
 *price\_percent\_change* = (*P2* - *P1*) / ((*P2* + *P1*) / 2)  
 return *quantity\_percent\_change* / *price\_percent\_change*# Для моделювання впливу субсидії необхідно врахувати, що субсидія зменшує витрати виробника, що призводить  
# до зміщення кривої пропозиції вліво на величину субсидії.  
  
# Введемо субсидію 0.5 до функції пропозиції, зменшивши ціну виробника  
def adjusted\_supply\_func(*p*, *subsidy*, *c*, *d*):  
 return supply\_func(*p* - *subsidy*, *c*, *d*)  
  
  
# Дані цін, попиту та пропозиції  
*prices* = [1, 1.25, 1.57, 1.81, 2.09, 2.45, 2.8, 3.19, 3.58, 3.85, 4.5, 5]  
*demand* = [280, 245, 190, 141, 135, 110, 95, 65, 58, 44, 21, 10]  
*supply* = [5, 20, 51, 89, 120, 153, 180, 201, 215, 228, 240, 248]  
  
# Створення DataFrame  
*market\_data* = pd.DataFrame({  
 'Price': *prices*,  
 'Demand': *demand*,  
 'Supply': *supply*})  
  
# Підбір параметрів для кривих попиту та пропозиції  
*popt\_demand*, *pcov\_demand* = curve\_fit(demand\_func, *market\_data*['Price'], *market\_data*['Demand'])  
*popt\_supply*, *pcov\_supply* = curve\_fit(supply\_func, *market\_data*['Price'], *market\_data*['Supply'])  
  
# Виведення параметрів  
*a\_demand*, *b\_demand* = *popt\_demand  
c\_supply*, *d\_supply* = *popt\_supply*# Построєння кривих попиту та пропозиції  
*price\_range* = np.linspace(min(*prices*), max(*prices*), 100)  
*demand\_curve* = demand\_func(*price\_range*, *a\_demand*, *b\_demand*)  
*supply\_curve* = supply\_func(*price\_range*, *c\_supply*, *d\_supply*)  
  
# Знаходження ціни та кількості в точці ринкової рівноваги  
*equilibrium\_price* = fsolve(equilibrium, 2, args=(*a\_demand*, *b\_demand*, *c\_supply*, *d\_supply*))[0]  
*equilibrium\_quantity* = supply\_func(*equilibrium\_price*, *c\_supply*, *d\_supply*)  
  
# Обчислення похідних у точці рівноваги  
*demand\_slope* = demand\_derivative(*equilibrium\_price*, *a\_demand*, *b\_demand*)  
*supply\_slope* = supply\_derivative(*equilibrium\_price*, *c\_supply*, *d\_supply*)  
  
# Виведення результатів  
print('Похідна кривої попиту ', *demand\_slope*)  
print('Похідна кривої пропозиції ', *supply\_slope*)  
print('Нестабільна ', *supply\_slope* > *demand\_slope*)  
  
# Отримаємо кількості та ціни для першого і останнього спостереження для попиту і пропозиції  
*Q1\_demand*, *Q2\_demand* = *market\_data*['Demand'].iloc[0], *market\_data*['Demand'].iloc[-1]  
*Q1\_supply*, *Q2\_supply* = *market\_data*['Supply'].iloc[0], *market\_data*['Supply'].iloc[-1]  
*P1*, *P2* = *market\_data*['Price'].iloc[0], *market\_data*['Price'].iloc[-1]  
  
# Обчислюємо дугову еластичність для попиту і пропозиції  
*arc\_elasticity\_demand* = arc\_elasticity(*Q1\_demand*, *Q2\_demand*, *P1*, *P2*)  
*arc\_elasticity\_supply* = arc\_elasticity(*Q1\_supply*, *Q2\_supply*, *P1*, *P2*)  
  
print('Дугова еластичність для попиту ', *arc\_elasticity\_demand*)  
print('Дугова еластичність для пропозиції ', *arc\_elasticity\_supply*)  
  
# Знаходження нової рівноважної ціни та кількості з урахуванням субсидії  
*subsidy* = 0.5  
*new\_equilibrium\_price* = fsolve(lambda *p*: equilibrium(*p*, *a\_demand*, *b\_demand*, *c\_supply*, *d\_supply*), *equilibrium\_price* + *subsidy*)[0]  
*new\_equilibrium\_quantity* = adjusted\_supply\_func(*new\_equilibrium\_price*, *subsidy*, *c\_supply*, *d\_supply*)  
  
# Визначимо ціни споживача та виробника  
*consumer\_price* = *new\_equilibrium\_price  
producer\_price* = *new\_equilibrium\_price* - *subsidy*print('Нова ціна споживача ', *consumer\_price*)  
print('Нова ціна виробника ', *producer\_price*)  
print('Нова точка рівноваги ', *new\_equilibrium\_quantity*)  
  
# Старий графік без врахуванням субсидії  
'''  
plt.figure(figsize=(10, 6))  
plt.scatter(market\_data['Price'], market\_data['Demand'], label='Demand data', color='blue')  
plt.scatter(market\_data['Price'], market\_data['Supply'], label='Supply data', color='red')  
plt.plot(price\_range, demand\_curve, label=f'Demand curve: $Q\_d = {a\_demand:.2f}P^{{-{b\_demand:.2f}}}$', color='blue')  
plt.plot(price\_range, supply\_curve, label=f'Supply curve: $Q\_s = {c\_supply:.2f}P^{{{d\_supply:.2f}}}$', color='red')  
plt.scatter(equilibrium\_price, equilibrium\_quantity, color='green', label='Equilibrium Point', zorder=5)  
plt.title('Market Demand and Supply Curves')  
plt.xlabel('Price')  
plt.ylabel('Quantity')  
plt.legend()  
plt.grid(True)  
plt.show()  
'''  
  
# Оновлений графік з врахуванням субсидії  
# Оновлений графік з врахуванням субсидії  
plt.figure(figsize=(10, 6))  
  
# Намалювати старі криві попиту та пропозиції  
plt.plot(*price\_range*, *demand\_curve*, label=f'Original Demand curve: $Q\_d = {*a\_demand*:.2f}P^{{-{*b\_demand*:.2f}}}$', color='blue')  
plt.plot(*price\_range*, *supply\_curve*, label=f'Original Supply curve: $Q\_s = {*c\_supply*:.2f}P^{{{*d\_supply*:.2f}}}$', color='red')  
  
# Намалювати нову криву пропозиції з урахуванням субсидії  
*new\_supply\_curve* = [adjusted\_supply\_func(*p*, *subsidy*, *c\_supply*, *d\_supply*) for *p* in *price\_range*]  
plt.plot(*price\_range*, *new\_supply\_curve*, label=f'Adjusted Supply curve with subsidy', color='orange', linestyle='--')  
  
# Відобразити стару та нову точки рівноваги  
plt.scatter(*equilibrium\_price*, *equilibrium\_quantity*, color='green', label='Original Equilibrium Point', zorder=5)  
plt.scatter(*consumer\_price*, *new\_equilibrium\_quantity*, color='purple', label='New Equilibrium Point', zorder=5)  
  
# Позначити ціни споживача та виробника  
plt.axhline(y=*new\_equilibrium\_quantity*, color='gray', linestyle='--', linewidth=0.5)  
plt.axvline(x=*consumer\_price*, color='purple', linestyle='--', linewidth=0.5, label='Consumer Price')  
plt.axvline(x=*producer\_price*, color='orange', linestyle='--', linewidth=0.5, label='Producer Price (after subsidy)')  
  
# Додати назви осей, заголовок, легенду та сітку  
plt.title('Market Demand and Supply Curves with Subsidy')  
plt.xlabel('Price')  
plt.ylabel('Quantity')  
plt.legend()  
plt.grid(True)  
plt.show()

**Висновок**

Я знайшов функції попиту та пропозиції, побудував їхні графіки. Також вийшло знайти точку рівноваги, еластичність попиту та пропозиції, та дугову еластичність. Були зроблені висновки з приводу стабільності рівноваги за допомогою програми. Також графічно відображена зміна параметрів ринкової рівноваги після введеня субсидії.