Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного автономного образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

ЛАБОРАТОРНАЯ РАБОТА

«АНАЛИТИЧЕСКИЙ И ЧИСЛЕННЫЙ (БРАУНА-РОБИНСОН) МЕТОДЫ РЕШЕНИЯ АНТАГОНИСТИЧЕСКОЙ ИГРЫ В СМЕШАННЫХ СТРАТЕГИЯХ»

по дисциплине: «*Методы принятия решений в программной инженерии*»

Выполнил: студент группы ИУК4-72Б _____ (Подпись) Губин Е.В. (И.О. Фамилия)

Проверил: _____ (Подпись) Никитенко У.В. (И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2025

Цель работы — изучить аналитический (обратной матрицы) и численный (Брауна — Робинсон) методы нахождения смешанных стратегий в антагонистической игре двух лиц в нормальной форме.

Постановка задачи

Найдите цену игры и оптимальные стратегии обоих игроков методами обратной матрицы и Брауна — Робинсон. Сравните полученные результаты.

$$\begin{pmatrix} 12 & 9 & 18 \\ 15 & 22 & 5 \\ 16 & 3 & 12 \end{pmatrix}$$

Рисунок 1 Входные данные

Результаты выполнения работы:

```
--- Аналитическое решение (метод обратной матрицы) ---
Цена игры v = 13.223076923077
Стратегия игрока I: [0.607692308, 0.346153846, 0.046153846]
Стратегия игрока II: [0.6, 0.130769231, 0.269230769]
```

Рисунок 2 Аналитической решение методом обратной матрицы

```
Запуск метода Брауна-Робинсон
итерация 1: нижн=5.000000, верхн=22.000000, разница=17.000000
итерация 2: нижн=5.000000, верхн=13.500000, разница=8.500000
итерация 3: нижн=9.333333, верхн=15.000000, разница=5.666667
итерация 4: нижн=11.500000, верхн=15.750000, разница=4.250000
итерация 5: нижн=12.800000, верхн=16.200000, разница=3.400000
итерация 6: нижн=13.000000, верхн=16.500000, разница=3.500000
итерация 7: нижн=12.714286, верхн=15.857143, разница=3.142857
итерация 8: нижн=12.250000, верхн=15.000000, разница=2.750000
итерация 9: нижн=11.888889, верхн=14.333333, разница=2.444444
итерация 10: нижн=11.600000, верхн=13.800000, разница=2.200000
итерация 1000: нижн=13.083000, верхн=13.332000, разница=0.249000
итерация 2000: нижн=13.175000, верхн=13.355500, разница=0.180500
итерация 3000: нижн=13.103333, верхн=13.245000, разница=0.141667
итерация 4000: нижн=13.207750, верхн=13.333750, разница=0.126000
итерация 5000: нижн=13.148600, верхн=13.259400, разница=0.110800
итерация 6000: нижн=13.197167, верхн=13.298167, разница=0.101000
итерация 7000: нижн=13.164143, верхн=13.255714, разница=0.091571
итерация 8000: нижн=13.146625, верхн=13.239750, разница=0.093125
итерация 9000: нижн=13.199778, верхн=13.298667, разница=0.098889
итерация 10000: нижн=13.203900, верхн=13.302900, разница=0.099000
итерация 11000: нижн=13.181818, верхн=13.276364, разница=0.094545
итерация 12000: нижн=13.199333, верхн=13.288500, разница=0.089167
итерация 13000: нижн=13.168923, верхн=13.252923, разница=0.084000
итерация 14000: нижн=13.177714, верхн=13.257857, разница=0.080143
итерация 15000: нижн=13.171067, верхн=13.246800, разница=0.075733
итерация 16000: нижн=13.157687, верхн=13.229875, разница=0.072188
итерация 17000: нижн=13.212941, верхн=13.282294, разница=0.069353
итерация 18000: нижн=13.187111, верхн=13.254667, разница=0.067556
итерация 19000: нижн=13.207579, верхн=13.272474, разница=0.064895
итерация 20000: нижн=13.195200, верхн=13.258350, разница=0.063150
```

Рисунок 3 Решение методом Брауна-Робинсона

```
--- Результат метода Брауна-Робинсон ---
Число итераций: 20000
Нижняя оценка: 13.195200
Верхняя оценка: 13.258350
Средняя оценка (mid): 13.226775
Средняя стратегия игрока I: [0.6153, 0.3438, 0.04095]
Средняя стратегия игрока II: [0.56405, 0.15085, 0.28515]
```

Рисунок 4 Результат решения методом Брауна-Робинсона

```
--- Сравнение аналитического и численного решений ---
Разность |v_mid - v_analytic| = 0.003698076923
Ошибка стратегии игрока I (L1): 0.015165384615
Ошибка стратегии игрока II (L1): 0.071950000000
```

Рисунок 5 Сравнение погрешностей решений

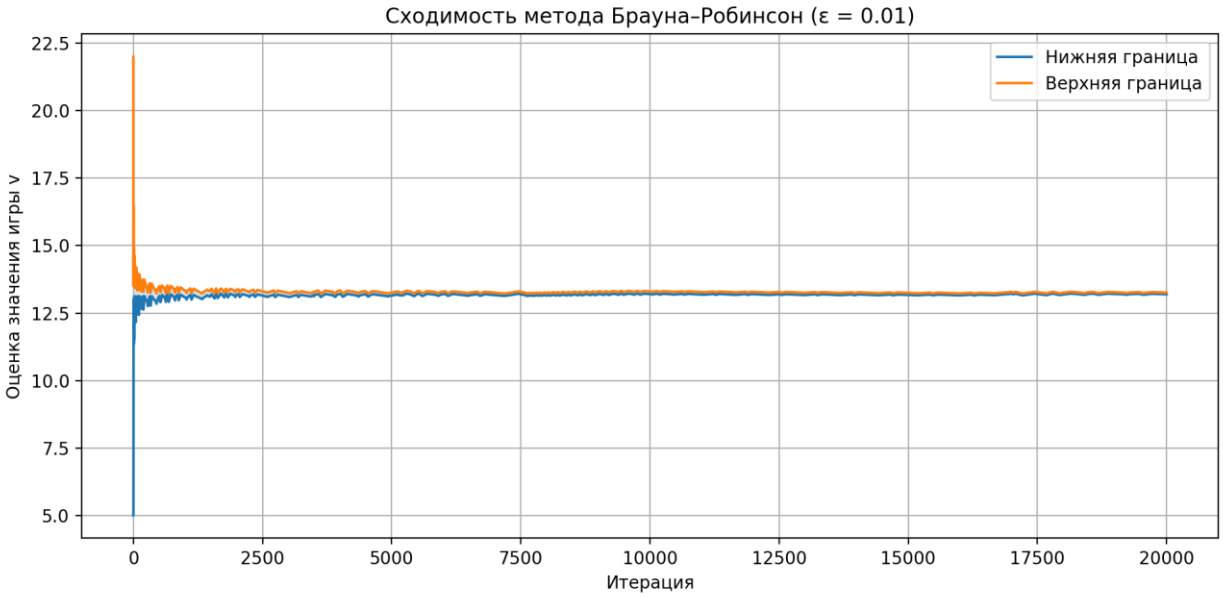

Рисунок 6 Сходимость решения методом Брауна-Робинсона

Листинг программы:

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

A = np.array([[12, 9, 18],
```

```
              [15, 22, 5],
              [16, 3, 12]], dtype=float)

EPS = 0.01
MAX_ITER = 20000
SAVE_PLOT = "br_convergence.png"
SAVE_CSV = "br_results_summary.csv"

def analytic_solution(A):
    n, m = A.shape
    if n != m:
        raise ValueError("Матрица должна быть квадратной для данного метода.")
    e = np.ones(n)
    det = np.linalg.det(A)
    if abs(det) < 1e-12:
        raise np.linalg.LinAlgError("Матрица вырождена — аналитическое решение
невозможно.")
    A_inv = np.linalg.inv(A)
    denom = float(e @ A_inv @ e)
    v = 1.0 / denom
    y = (A_inv @ e) / denom
    A_T_inv = np.linalg.inv(A.T)
    denom2 = float(e @ A_T_inv @ e)
    x = (A_T_inv @ e) / denom2
    return v, x, y

def brown_robinson(A, eps=0.01, max_iter=20000, verbose=False):
    m, n = A.shape
    row_counts = np.zeros(m)
    col_counts = np.zeros(n)
    init_row = int(np.argmax(A @ (np.ones(n) / n)))
    init_col = int(np.argmin((np.ones(m) / m) @ A))
    row_counts[init_row] += 1
    col_counts[init_col] += 1
    values_lower = []
    values_upper = []
    history = []
    for t in range(1, max_iter + 1):
        x_avg = row_counts / t
        y_avg = col_counts / t
        col_pay = x_avg @ A
        v_lower = float(np.min(col_pay))
        row_pay = A @ y_avg
        v_upper = float(np.max(row_pay))
        values_lower.append(v_lower)
        values_upper.append(v_upper)
        history.append({
            'iter': t,
            'x_avg': x_avg.copy(),
            'y_avg': y_avg.copy(),
            'v_lower': v_lower,
            'v_upper': v_upper
        })
        if verbose and (t <= 10 or t % 1000 == 0):
              print(f"итерация {t}: нижн={v_lower:.6f}, верхн={v_upper:.6f},
разница={v_upper - v_lower:.6f}")
```

```
        if (v_upper - v_lower) <= eps and t > 1:
            break
        next_row = int(np.argmax(A @ y_avg))
        next_col = int(np.argmin(x_avg @ A))
        row_counts[next_row] += 1
        col_counts[next_col] += 1
    return {
        'iterations': t,
        'x_avg': row_counts / t,
        'y_avg': col_counts / t,
        'v_lower': v_lower,
        'v_upper': v_upper,
        'values_lower': np.array(values_lower),
        'values_upper': np.array(values_upper),
        'history': history
    }

def print_solution(v, x, y, title="Аналитическое решение"):
    print(f"\n--- {title} ---")
    print(f"Цена игры v = {v:.12f}")
    print(f"Стратегия игрока I: {np.round(x, 9).tolist()}")
    print(f"Стратегия игрока II: {np.round(y, 9).tolist()}\n")

def save_summary_csv(filename, analytic_v, analytic_x, analytic_y, br_res):
    rows = []
    rows.append({'name': 'v_analytic', 'value': analytic_v})
    for i, xi in enumerate(analytic_x, start=1):
        rows.append({'name': f'x_analytic_{i}', 'value': xi})
    for j, yj in enumerate(analytic_y, start=1):
        rows.append({'name': f'y_analytic_{j}', 'value': yj})
    rows.append({'name': 'BR_iterations', 'value': br_res['iterations']})
    for i, xi in enumerate(br_res['x_avg'], start=1):
        rows.append({'name': f'BR_x{i}', 'value': xi})
    for j, yj in enumerate(br_res['y_avg'], start=1):
        rows.append({'name': f'BR_y{j}', 'value': yj})
    rows.append({'name': 'BR_v_lower', 'value': br_res['v_lower']})
    rows.append({'name': 'BR_v_upper', 'value': br_res['v_upper']})
    pd.DataFrame(rows).to_csv(filename, index=False)
    print(f"Сводная таблица сохранена в {filename}")

if __name__ == "__main__":
    np.set_printoptions(precision=9, suppress=True)
    try:
        v_a, x_a, y_a = analytic_solution(A)
         print_solution(v_a, x_a, y_a, "Аналитическое решение (метод обратной
матрицы)")
    except Exception as e:
        print("Ошибка аналитического метода:", e)
        v_a, x_a, y_a = None, None, None
    print("Запуск метода Брауна-Робинсон")
    br = brown_robinson(A, eps=EPS, max_iter=MAX_ITER, verbose=True)
    print("\n--- Результат метода Брауна-Робинсон ---")
    print(f"Число итераций: {br['iterations']}")
    print(f"Нижняя оценка: {br['v_lower']:.6f}")
    print(f"Верхняя оценка: {br['v_upper']:.6f}")
    print(f"Средняя оценка (mid): {(br['v_lower'] + br['v_upper']) / 2:.6f}")
```

```
    print(f"Средняя стратегия игрока I: {np.round(br['x_avg'], 6).tolist()}")
         print(f"Средняя   стратегия   игрока   II:   {np.round(br['y_avg'],
6).tolist()}\n")
    if v_a is not None:
        v_mid = (br['v_lower'] + br['v_upper']) / 2
        print("--- Сравнение аналитического и численного решений ---")
        print(f"Разность |v_mid - v_analytic| = {abs(v_mid - v_a):.12f}")
         print(f"Ошибка стратегии игрока I (L1): {np.sum(np.abs(br['x_avg'] -
x_a)):.12f}")
         print(f"Ошибка стратегии игрока II (L1): {np.sum(np.abs(br['y_avg'] -
y_a)):.12f}\n")
    save_summary_csv(SAVE_CSV, v_a, x_a, y_a, br)
    vals_low = br['values_lower']
    vals_up = br['values_upper']
    iters = np.arange(1, len(vals_low) + 1)
    plt.figure(figsize=(10, 5))
    plt.plot(iters, vals_low, label='Нижняя граница')
    plt.plot(iters, vals_up, label='Верхняя граница')
    plt.fill_between(iters, vals_low, vals_up, alpha=0.2)
    plt.xlabel('Итерация')
    plt.ylabel('Оценка значения игры v')
    plt.title(f'Сходимость метода Брауна-Робинсон (ε = {EPS})')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.savefig(SAVE_PLOT, dpi=200)
    print(f"График сохранён в {SAVE_PLOT}")
    last = br['history'][-20:]
    rows = []
    for rec in last:
        rows.append({
            'iter': rec['iter'],
            'v_lower': rec['v_lower'],
            'v_upper': rec['v_upper'],
            'x1': rec['x_avg'][0],
            'x2': rec['x_avg'][1],
            'x3': rec['x_avg'][2],
            'y1': rec['y_avg'][0],
            'y2': rec['y_avg'][1],
            'y3': rec['y_avg'][2],
        })
    pd.DataFrame(rows).to_csv("br_last_iterations.csv", index=False)
    print("Таблица последних итераций сохранена: br_last_iterations.csv")
    print("\nГотово!")
```

Вывод: в ходе выполнения лабораторной работы были получены практические навыки по программному поиску оптимальных стратегий методами обратной матрицы и Брауна — Робинсон.