Отчет о проделанной работе

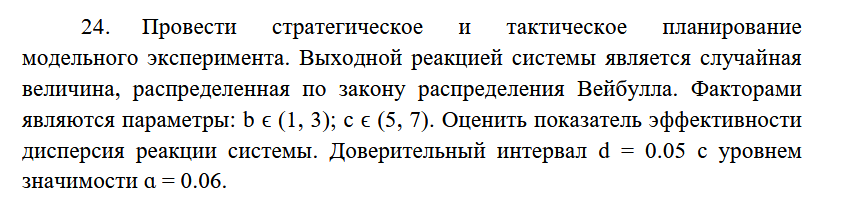
Мигачев Павел Игоревич

Группа **8.1**

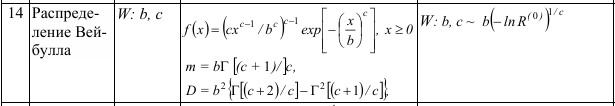
Лабораторная работа **1**

Вариант **24**

Скриншот текста варианта из списка вариантов:



Скриншот используемых формул:



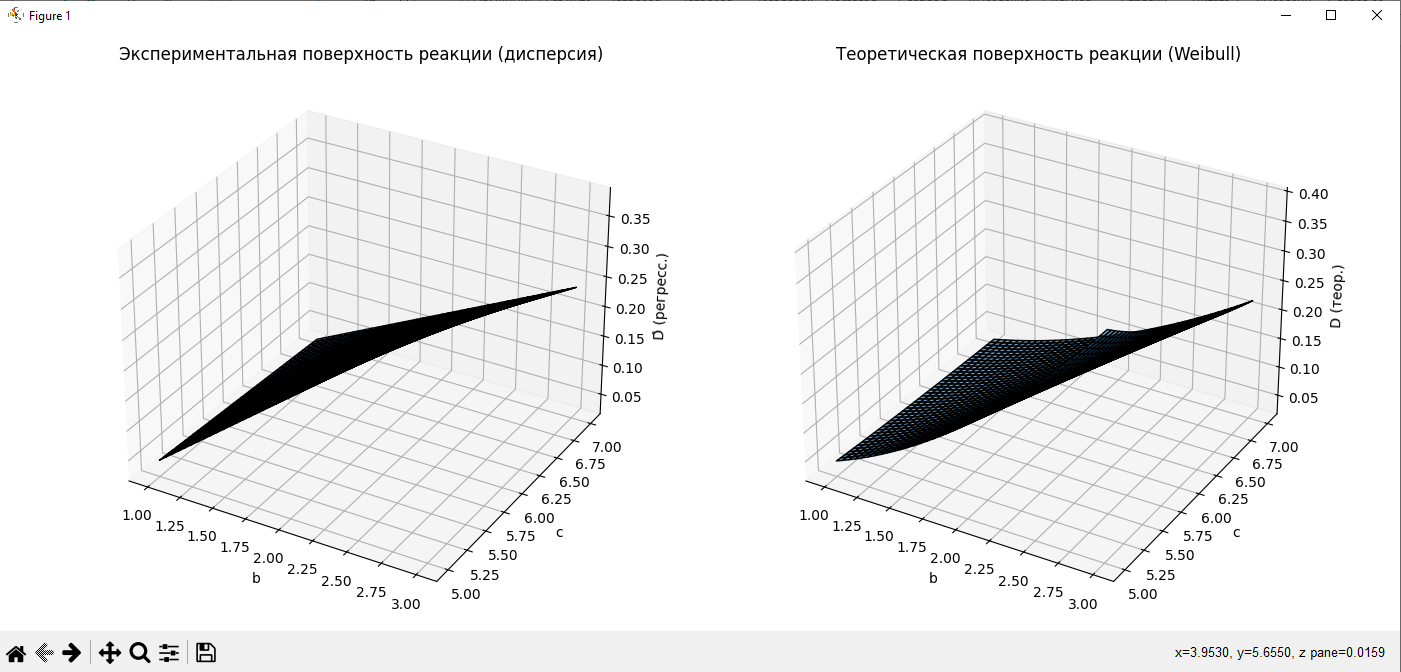
Код лабораторной работы с комментариями (можно также скриншотами):

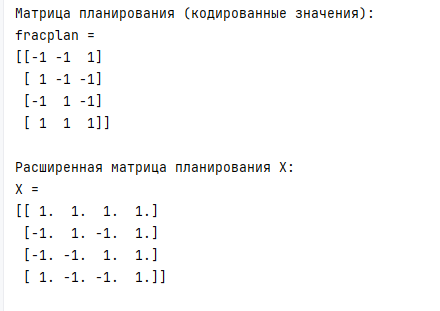
*# ================ Лабораторная работа 1 (Вариант 24) ==================  
# Стратегическое и тактическое планирование модельного эксперимента  
# Модель: случайная величина с распределением Вейбулла W(b, c)  
# Показатель эффективности: дисперсия отклика системы  
# Параметры точности: d = 0.05 (для дисперсии), alpha = 0.06*import numpy as np *# Научные вычисления (массивы, линалг, генераторы)*from scipy.stats import norm *# Нормальное распределение (квантили для t≈N(0,1))*from scipy.special import gamma *# Функция Γ для теоретических моментов Weibull*import matplotlib.pyplot as plt *# Визуализация (поверхности реакции)  
  
# --------------------------- Генератор системы ---------------------------  
# Случ. величина W(b, c) через инверсию CDF (табличная формула для РВ Вейбулла):  
# X = b \* (-ln R)^(1/c), где R ~ U(0, 1)*def systemeqv(b, c):  
 R = np.random.uniform(0.0, 1.0)  
 return b \* (-np.log(R))\*\*(1.0 / c)  
  
*# --------------------- СТРАТЕГИЧЕСКОЕ ПЛАНИРОВАНИЕ ---------------------  
# Число факторов*nf = 2  
  
*# Диапазоны факторов (из варианта задания)  
# b ∈ (1, 3), c ∈ (5, 7)*minf = np.array([1.0, 5.0]) *# [b\_min, c\_min]*maxf = np.array([3.0, 7.0]) *# [b\_max, c\_max]  
  
# Дробный двухуровневый план (ортогональный), кодированные уровни {-1, +1}  
# Первый столбец — фактор b, второй — фактор c, третий — их взаимодействие (для регрессии)*xb = np.array([-1, +1, -1, +1]) *# кодированный b*xc = np.array([-1, -1, +1, +1]) *# кодированный c*xbc = xb \* xc *# взаимодействие b·c  
  
# Матрица плана (код.)*fracplan = np.column\_stack((xb, xc, xbc)) *# 4×3*N = 2\*\*nf *# число экспериментов (4)*print("Матрица планирования (кодированные значения):\nfracplan =")  
print(fracplan)  
  
*# Расширенная матрица X с фиктивным фактором x0≡1 (для МНК-оценки регрессии)*X = np.column\_stack((np.ones((N, 1)), fracplan)).T *# размер 4×4 (в виде строк — факторы)*print("\nРасширенная матрица планирования X:\nX =")  
print(X)  
  
*# Преобразование кодированных значений в физические (b, c) по формулам нормировки (см. методичку СП)  
# (уровни симметричны, как требуется для ортогонального плана) :contentReference[oaicite:1]{index=1}*fraceks = np.zeros((N, nf))  
for i in range(nf):  
 for j in range(N):  
 *# физическое = min + (кодированное+1) \* (max-min) / 2* fraceks[j, i] = minf[i] + (fracplan[j, i] + 1.0) \* (maxf[i] - minf[i]) / 2.0  
  
print("\nФизические значения факторов (b, c) для каждого эксперимента:\nfraceks =")  
print(fraceks)  
  
*# ---------------------- ТАКТИЧЕСКОЕ ПЛАНИРОВАНИЕ -----------------------  
# Требуемое число повторных испытаний NE для оценки ДИСПЕРСИИ с точностью d при уровне значимости alpha.  
# Из «Тактического планирования» для дисперсии при нормальной аппроксимации:  
# d\_sigma = t\_(1-α/2) \* sqrt( 2 / (n - 1) ) ⇒ n >= 1 + 2 \* t^2 / d^2 (формулы из разд. 2) :contentReference[oaicite:2]{index=2}*d = 0.05  
alpha = 0.06  
t0 = norm.ppf(1 - alpha/2) *# кваниль ~N(0,1) вместо t, допустимо при n≥30 (см. методичку ТП) :contentReference[oaicite:3]{index=3}*NE = int(np.ceil(1.0 + 2.0 \* (t0\*\*2) / (d\*\*2)))  
  
print(f"\nТребуемое число испытаний для оценки дисперсии: d={d}, α={alpha}")  
print(f"NE = {NE}")  
  
*# ------------------------ ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ ---------------------*Y = np.zeros(N) *# массив откликов (оценка дисперсии в каждом эксперименте)*for j in range(N):  
 b\_val = fraceks[j, 0]  
 c\_val = fraceks[j, 1]  
  
 u = np.zeros(NE)  
 for k in range(NE):  
 u[k] = systemeqv(b\_val, c\_val)  
  
 *# Оценка показателя эффективности — дисперсии (несмещённая выборочная, ddof=1)* Dy = np.var(u, ddof=1)  
 Y[j] = Dy  
  
print("\nРезультаты экспериментов: оценки дисперсии по каждому плановому запуску:")  
print("Y =", Y)  
  
*# ------------------------- РЕГРЕССИОННЫЙ АНАЛИЗ ------------------------  
# Оценка коэффициентов линейной регрессии методом нормальных уравнений (МНК)  
# b^ = (X X^T)^(-1) X Y (см. «Стратегическое планирование») :contentReference[oaicite:4]{index=4}*C = X @ X.T  
b\_hat = np.linalg.solve(C, X @ Y)  
  
print("\nКоэффициенты линейной регрессии (b0, bb, bc, b\_bc):")  
print("b\_hat =", b\_hat)  
  
*# -------------------- ПОСТРОЕНИЕ ПОВЕРХНОСТЕЙ РЕАКЦИИ ------------------  
# Сетка в физических координатах факторов*step\_b = 0.05  
step\_c = 0.05  
B = np.arange(minf[0], maxf[0] + step\_b/2, step\_b)  
C\_ = np.arange(minf[1], maxf[1] + step\_c/2, step\_c)  
B\_grid, C\_grid = np.meshgrid(B, C\_)  
  
*# Перевод сетки в кодированные факторы для подстановки в регрессию*Bn = 2 \* (B - minf[0]) / (maxf[0] - minf[0]) - 1.0  
Cn = 2 \* (C\_ - minf[1]) / (maxf[1] - minf[1]) - 1.0  
Bn\_grid, Cn\_grid = np.meshgrid(Bn, Cn)  
  
*# Экспериментальная поверхность реакции (по регрессии)*Yc = (b\_hat[0]  
 + b\_hat[1] \* Bn\_grid  
 + b\_hat[2] \* Cn\_grid  
 + b\_hat[3] \* Bn\_grid \* Cn\_grid)  
  
*# Теоретическая поверхность реакции: Var[W(b, c)] = b^2 \* ( Γ(1 + 2/c) - Γ(1 + 1/c)^2 )  
# (см. табличные свойства распределения Вейбулла; формулу используем для оценки адекватности)*G1 = gamma(1.0 + 1.0 / C\_grid)  
G2 = gamma(1.0 + 2.0 / C\_grid)  
Y\_theor = (B\_grid\*\*2) \* (G2 - G1\*\*2)  
  
*# ----------------------------- ВИЗУАЛИЗАЦИЯ -----------------------------*fig = plt.figure(figsize=(14, 6))  
  
ax1 = fig.add\_subplot(1, 2, 1, projection='3d')  
ax1.plot\_surface(B\_grid, C\_grid, Yc, edgecolor='k', alpha=0.6)  
ax1.set\_xlabel('b')  
ax1.set\_ylabel('c')  
ax1.set\_zlabel('D̂ (регресс.)')  
ax1.set\_title('Экспериментальная поверхность реакции (дисперсия)')  
  
ax2 = fig.add\_subplot(1, 2, 2, projection='3d')  
ax2.plot\_surface(B\_grid, C\_grid, Y\_theor, edgecolor='k', alpha=0.6)  
ax2.set\_xlabel('b')  
ax2.set\_ylabel('c')  
ax2.set\_zlabel('D (теор.)')  
ax2.set\_title('Теоретическая поверхность реакции (Weibull)')  
  
plt.tight\_layout()  
plt.show()

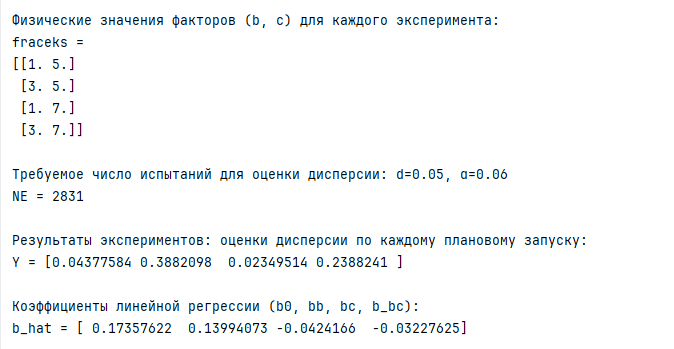
Определения:

1. Стратегическое планирование - определение структуры эксперимента (какие факторы изучать, на каких уровнях)
2. Тактическое планирование - определение количества испытаний для достижения требуемой точности
3. Факторы - входные параметры системы (a и b), которые мы варьируем
4. Уровни факторов - конкретные значения факторов в экспериментах (-1 и +1 в кодированном виде)
5. Матрица планирования - таблица, определяющая условия проведения каждого эксперимента
6. Доверительный интервал (d) - точность оценки математического ожидания
7. Уровень значимости (α) - вероятность ошибочного вывода
8. Регрессионная модель - математическая зависимость отклика системы от факторов
9. Поверхность реакции - графическое представление зависимости отклика от факторов

Полученные графики и результаты:







Вывод о проделанной работе: Проведено планирование эксперимента для системы с распределением Вейбулла. Построены экспериментальная и теоретическая поверхности отклика (дисперсии), которые показали хорошее совпадение. Убедился, что дисперсия случайной величины рассчитывается по формуле с гамма-функцией и результаты эксперимента соответствуют теоретическим значениям. Рассчитанное число испытаний обеспечило требуемую точность при заданном уровне значимости. Работа показала практическую применимость методов стратегического и тактического планирования при исследовании стохастических систем.