Поповкин Артемий Андреевич
Б9122-02.03.01 СЦТ

Постановка задачи

Требуется для **модели Брусселятора** исследовать динамику двух связанных переменных, описывающих концентрации реагентов в модели автоколебательной химической реакции, и проанализировать переход системы от устойчивого состояния к колебательному режиму в зависимости от параметров системы и начальных условий

Описание модели

Описание

Модель Брюсселятора – динамическая система, предложенная И. Пригожиным и Р. Лефевром как упрощенная модель химической реакции типа "реакция-диффузия".

Обозначения

- t — время

Переменные и параметры

- A, B — исходные вещества
- X, Y — промежуточные вещества
- D, E — конечные продукты
- k_i — константы скоростей реакций

Уравнение

Имеем схему химических реакций

$$\begin{cases} A \xrightarrow{k_1} X \\ B + X \xrightarrow{k_2} Y + D \\ 2X + Y \xrightarrow{k_3} 3X \\ X \xrightarrow{k_4} E \end{cases}$$

Исходная модель имеет вид:

$$\begin{cases} \dfrac{dX}{dt} = k_1 A - k_2 BX + k_3 X^2 Y - k_4 X \\ \dfrac{dY}{dt} = k_2 BX - k_3 X^2 Y \end{cases}$$

Введём безразмерные переменные и параметры:

$$x = \sqrt{\frac{k_3}{k_4}} X \quad y = \sqrt{\frac{k_3}{k_4}} Y, \quad \tau = k_4 t, \quad \rho = r\sqrt{\frac{k_4}{Dx}}$$

$$a = \sqrt{\frac{k_3}{k_4}} \cdot \frac{k_1}{k_4} A, \quad b = \frac{k_2}{k_4} B$$

Подставляя в исходную систему, получаем безразмерную форму модели Брюсселятора

$$\begin{cases} \dot{x} = a - (b+1)x + x^2 y \\ \dot{y} = bx - x^2 y \end{cases}$$

Где $\dot{}$ означает производную по безразмерному времени τ, а индекс ρ — производную по безразмерной пространственной координате ρ

Аналитическое исследование системы

Условие устойчивости

Для $D_X = D_Y = 0$ найдём равновесия, приравнивая правые части к 0:

$$\begin{cases} a - (b+1)x + x^2 y = 0 \\ bx - x^2 y = 0 \end{cases}$$

Решив систему, получаем координаты единственного нетривиального положения равновесия

Виды равновесий

Нетривиальное

$x^\star = a, \quad y^\star = \dfrac{b}{a}$

Исследование на устойчивость

Найдём матрицу Якоби системы

$$J(x, y) = \begin{pmatrix} -(b+1) + 2xy & x^2 \\ b - 2xy & -x^2 \end{pmatrix}$$

Подставляем точку равновесия

$$J\left(a, \frac{b}{a}\right) = \begin{pmatrix} b-1 & a^2 \\ -b & -a^2 \end{pmatrix}$$

Найдём собственные значения матрицы
$\lambda^2 + \lambda\left(a^2 - b + 1\right) + a^2 = 0$

Воспользуемся критерием Гурвица:

$$\begin{cases} a^2 > 0 \\ a^2 - b + 1 > 0 \end{cases}$$

Данное условие эквивалентно $b < a^2 + 1$

Исследование на устойчивость с учётом диффузии

Рассмотрим систему с диффузией

$$\dot{x} = a - (b+1)x + x^2y + D_x x_{rr} \dot{y} = bx - x^2y + D_y y_{rr}$$

$$\xi\left(r,t\right) = x\left(r,t\right) - x_0, \quad \eta\left(r,t\right) = y\left(r,t\right) - y_0$$

Линеаризуя систему в окрестности точки (x_0, y_0) получим

$$\begin{cases} \dot{\xi} = (b-1)\xi + a^2\eta + D_x\xi_{rr} \\ \dot{\eta} = -b\xi - a^2\eta + D_y\eta_{rr} \end{cases}$$

Решение линеаризованной системы ищем в виде плоских волн

$$\begin{cases} \xi\left(r,t\right) = C_1 e^{pt+ikr} \\ \eta\left(r,t\right) = C_2 e^{pt+ikr} \end{cases}$$

где p - показатель роста возмущения (a. k. a. мода), k - волновое число.
Подставляя это в линеаризованную систему, получаем систему линейных уравнений для амплитуд C_1, C_2

$$\begin{cases} \left(b - 1 - k^2D_x - p\right)C_1 + a^2C_2 = 0 \\ -bC_1 + \left(-a^2 - D_yk^2 - p\right)C_2 = 0 \end{cases}$$

Нетривиальное решение существует при условии равенства нулю определителя

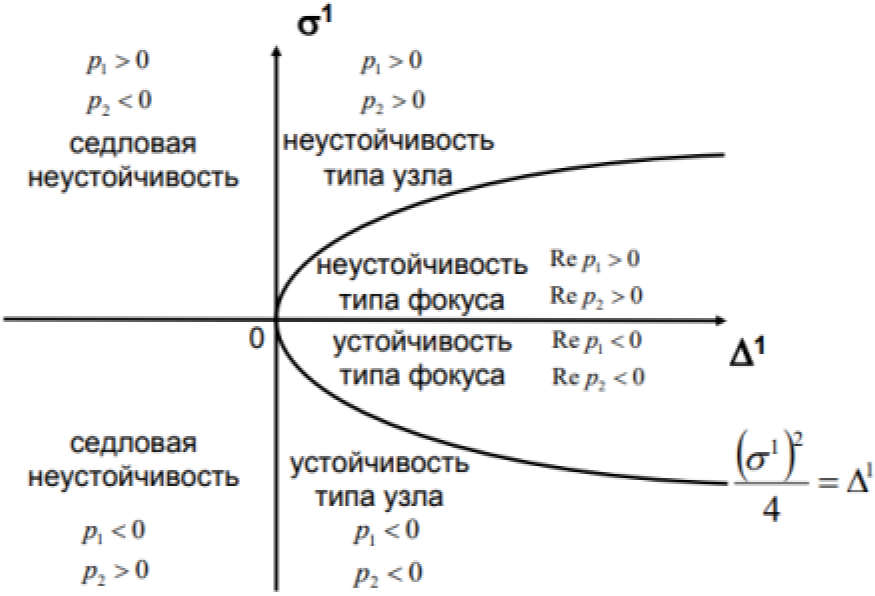
$$\begin{vmatrix} b - 1 - D_xk^2 - p & a^2 \\ -b & -a^2 - D_yk^2 - p \end{vmatrix} = 0$$

Раскрывая определитель, получаем дисперсионное уравнение

$$p^2 + \sigma^1 p + \Delta^1 = 0$$

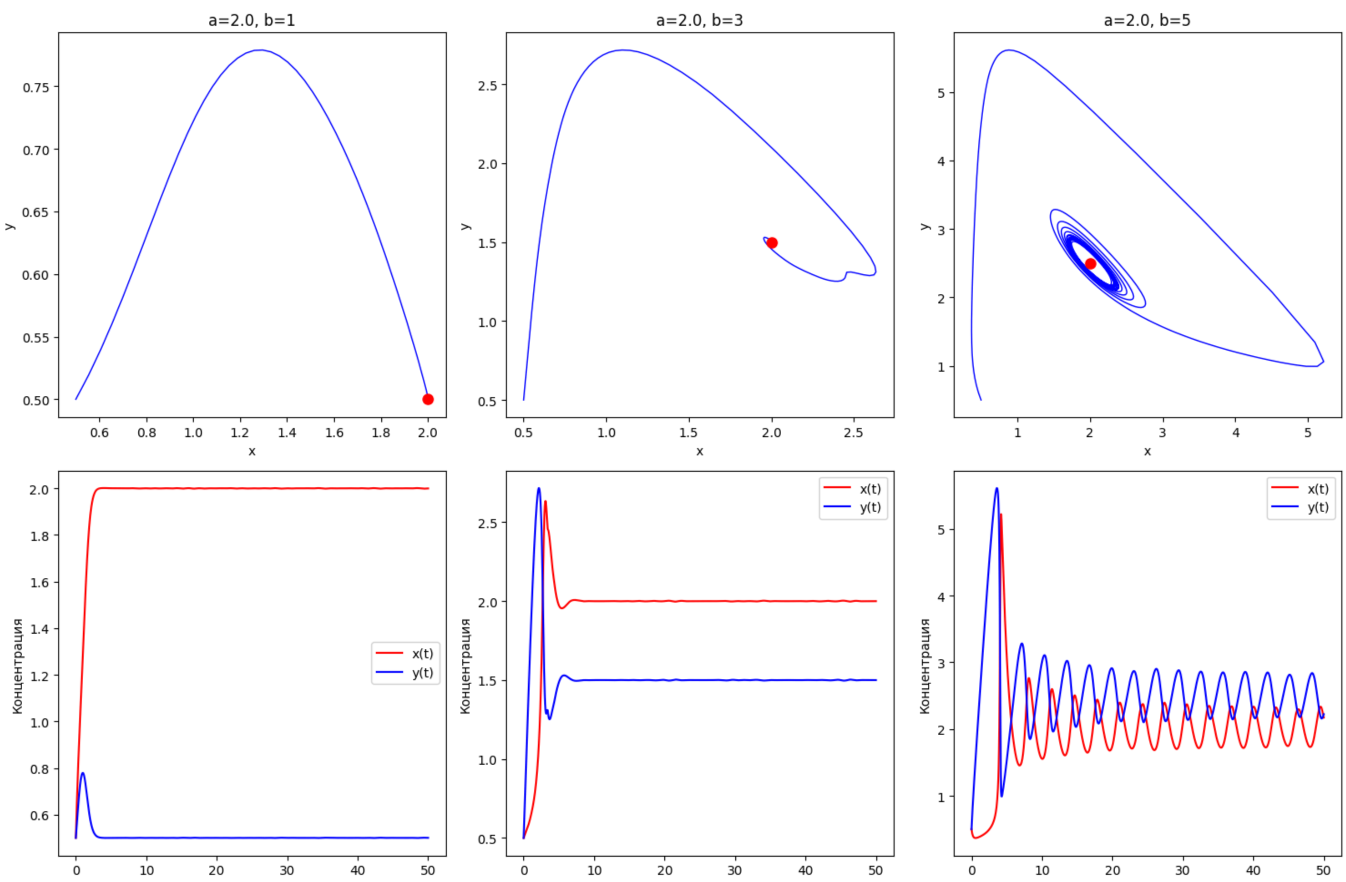
$\sigma^1 = (D_x + Dy)k^2 + a^2 - b + 1, \quad \Delta^1 = D_xD_yk^4 + \left(a^2Dx - (b-1)Dy\right)k^2 + a^2$

С учётом этого получаем



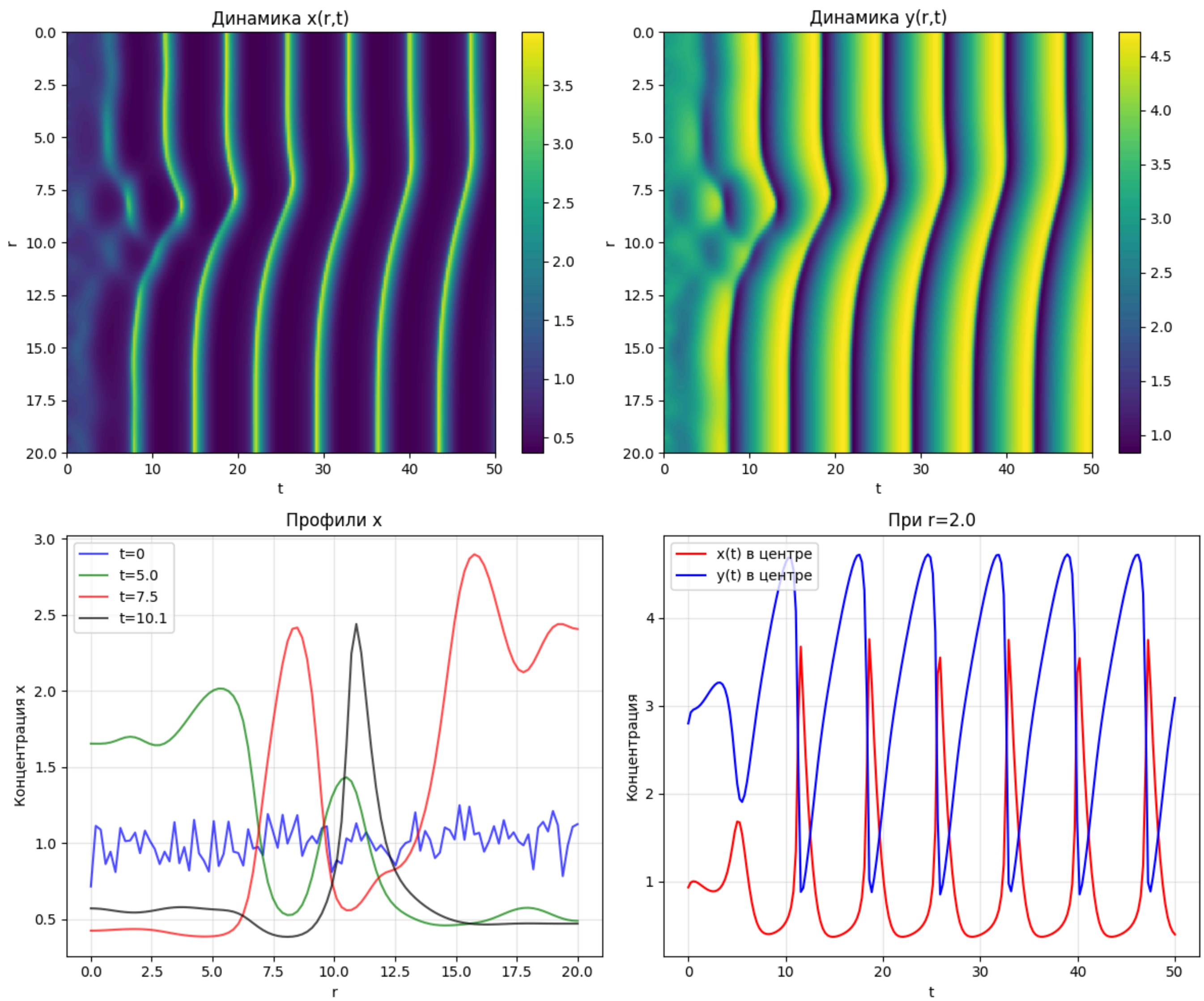
Численное исследование модели

В рамках численного исследования модели, для ситуации без учёта диффузии с помощью численных методов семейства Рунге-Кутта 4-го порядка были построены фазовые портреты и графики динамики модели при различных значениях параметров



Можем видеть, что в модели при сравнительно малом b наблюдается стационарная динамика.
При $b = a^2 + 1$ действительно происходит бифуркация, а при дальнейшем увеличении b модель демонстрирует колебательный режим.

Далее рассмотрим случаи, когда вещества могут диффундировать ($D_x \neq 0,\ D_y \neq 0$)
Для значений параметров $D_x = 0.1,\ D_y = 0.05,\ a = 1,\ b = 3$ методом конечных элементов было получено численное решение модели на временном интервале $T = [0, 50]$
Следующие графики демонстрируют динамику модели в различных плоскостях:



Можем видеть, что при этих значениях параметров, модель с течением времени сходится к периодической динамике.
В каждой фиксированной точке пространства r мы наблюдаем периодически меняющиеся концентрации. В пространстве в целом, после стабилизации, наблюдаются периодически возникающие бегущие волны, распространяющиеся в сторону увеличения r с постоянной скоростью.

Заключение

В ходе исследования была изучена модель Брюсселятора - классическая модель для описания самоорганизации в химических реакциях.
Проведенный анализ показал, что система переходит от устойчивого состояния к автоколебаниям при изменении бифуркационного параметра.
Численное моделирование подтвердило теоретические результаты и позволило визуализировать динамику системы.
Результаты работы демонстрируют принципы самоорганизации, показывая, как из простых взаимодействий могут возникать сложные пространственно-временные структуры.

Листинг кода

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
def brusselator_ode(t, u, a, b):
    x, y = u
    dxdt = a - (b + 1) * x + x**2 * y
    dydt = b * x - x**2 * y
    return [dxdt, dydt]


params = [{"a": 2.0, "b": 3}, {"a": 2.0, "b": 5}, {"a": 2.0, "b": 7}]
fig, axes = plt.subplots(2, 3, figsize=(15, 10))
for i, (param) in enumerate(params):
    a, b = param["a"], param["b"]
    # Решение системы
    t_span = [0, 50]
    t_eval = np.linspace(0, 50, 1000)
    sol = solve_ivp(
        brusselator_ode, t_span, [0.5, 0.5], args=(a, b), t_eval=t_eval,
method="RK45"
    )
    # Фазовый портрет
    axes[0, i].plot(sol.y[0], sol.y[1], "b-", linewidth=1)
    axes[0, i].plot(a, b / a, "ro", markersize=8)  # Равновесие
    axes[0, i].set_xlabel("x")
    axes[0, i].set_ylabel("y")
    axes[0, i].set_title(f"a={a}, b={b}")
    # axes[0, i].grid(True)
    # Временные ряды
    axes[1, i].plot(sol.t, sol.y[0], "r-", label="x(t)")
    axes[1, i].plot(sol.t, sol.y[1], "b-", label="y(t)")
    axes[1, i].set_xlabel("Время")
```

```
    axes[1, i].set_ylabel("Концентрация")
    axes[1, i].legend()
    # axes[1, i].grid(True)
plt.tight_layout()
plt.show()
# Параметры для модели с диффузией
Dx = 0.1
Dy = 0.05
L = 20.0
N = 100
dx = L / (N - 1)
# Начальные условия с небольшим пространственным возмущением
x0 = a + 0.1 * np.random.randn(N)
y0 = b / a + 0.1 * np.random.randn(N)


# Функция для модели с диффузией
def brusselator_diffusion(t, state):
    x = state[:N]
    y = state[N:]
    dxdt = np.zeros_like(x)
    dydt = np.zeros_like(y)
    for i in range(1, N - 1):
        dxdt[i] = (
            a
            - (b + 1) * x[i]
            + x[i] ** 2 * y[i]
            + Dx * (x[i + 1] - 2 * x[i] + x[i - 1]) / dx**2
        )
        dydt[i] = (
            b * x[i] - x[i] ** 2 * y[i] + Dy * (y[i + 1] - 2 * y[i] + y[i -
1]) / dx**2
        )
    dxdt[0] = a - (b + 1) * x[0] + x[0] ** 2 * y[0] + Dx * 2 * (x[1] - x[0]) /
dx**2
    dxdt[-1] = (
        a - (b + 1) * x[-1] + x[-1] ** 2 * y[-1] + Dx * 2 * (x[-2] - x[-1]) /
dx**2
    )
    dydt[0] = b * x[0] - x[0] ** 2 * y[0] + Dy * 2 * (y[1] - y[0]) / dx**2
    dydt[-1] = b * x[-1] - x[-1] ** 2 * y[-1] + Dy * 2 * (y[-2] - y[-1]) /
dx**2
    return np.concatenate([dxdt, dydt])


# Решение системы с диффузией
```

```
initial_state_diff = np.concatenate([x0, y0])
t_span_diff = (0, 50)
t_eval_diff = np.linspace(t_span_diff[0], t_span_diff[1], 200)
solution_diff = solve_ivp(
    brusselator_diffusion,
    t_span_diff,
    initial_state_diff,
    t_eval=t_eval_diff,
    method="BDF",
    rtol=1e-6,
)
# Визуализация пространственно-временной динамики
x_space_time = solution_diff.y[:N, :]
y_space_time = solution_diff.y[N:, :]
space = np.linspace(0, L, N)
time = solution_diff.t
fig, axes = plt.subplots(2, 2, figsize=(12, 10))
# Пространственно-временная диаграмма для x
im1 = axes[0, 0].imshow(x_space_time, aspect="auto", extent=[0, time[-1], L,
0])
axes[0, 0].set_xlabel("t")
axes[0, 0].set_ylabel("r")
axes[0, 0].set_title("Динамика x(r,t)")
plt.colorbar(im1, ax=axes[0, 0])
# Пространственно-временная диаграмма для y
im2 = axes[0, 1].imshow(y_space_time, aspect="auto", extent=[0, time[-1], L,
0])
axes[0, 1].set_xlabel("t")
axes[0, 1].set_ylabel("r")
axes[0, 1].set_title("Динамика y(r,t)")
plt.colorbar(im2, ax=axes[0, 1])
# Профили концентраций в разные моменты времени
axes[1, 0].plot(space, x_space_time[:, 0], "b-", label="t=0", alpha=0.7)
axes[1, 0].plot(space, x_space_time[:, 20], "g-", label=f"t={time[20]:.1f}",
alpha=0.7)
axes[1, 0].plot(space, x_space_time[:, 30], "r-", label=f"t={time[30]:.1f}",
alpha=0.7)
axes[1, 0].plot(space, x_space_time[:, 40], "k-", label=f"t={time[40]:.1f}",
alpha=0.7)
axes[1, 0].set_xlabel("r")
axes[1, 0].set_ylabel("Концентрация x")
axes[1, 0].set_title("Профили x")
axes[1, 0].legend()
axes[1, 0].grid(True, alpha=0.3)
# Осцилляции в одной точке пространства
point_idx = 10
```

```
axes[1, 1].plot(time, x_space_time[point_idx, :], "r-", label="x(t) в центре")
axes[1, 1].plot(time, y_space_time[point_idx, :], "b-", label="y(t) в центре")
axes[1, 1].set_xlabel("t")
axes[1, 1].set_ylabel("Концентрация")
axes[1, 1].set_title(f"При r={space[point_idx]:.1f}")
axes[1, 1].legend()
axes[1, 1].grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```