

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ФРАНКА
МЕХАНІКО-МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра математичної
економіки, економетрії,
фінансової та страхової
математики

МАГІСТЕРСЬКА РОБОТА

”Чисельна реалізація оптимального
керування динамічною моделлю популяції”

Виконала: студентка групи МТЕМ-61с
спеціальності 111 - математика,
спеціалізації математична економіка та
економетрика

Цвик Христина Романівна

Науковий керівник :
доц. Флюд В.М.

Роботу рекомендовано до захисту
на засіданні кафедри математичної
економіки, економетрії, фінансової та
страхової математики

протокол від 04 грудня 2020 року №4

В. о. завідувача кафедрою
проф. Оліскевич М. О.

Зміст

1	Вступ	2
2	Дифузія в математичних моделях	3
3	Постановка задачі	6
4	Існування оптимального контролю	10
5	Принцип максимуму	13
6	Градiєнтний метод	17
7	Алгоритм градієнта для проблеми (*)	18
8	Реалізація задачі у пакеті Python	20
9	Отримані результати	28
10	Висновки	31
11	Використана література	32

1 Вступ

Теорія управління швидко розвивалася з моменту появи перших робіт Понтрягіна та співавторів наприкінці 1950-х років, і зараз вона визнана як важлива область прикладної математики. Теорія оптимального управління та оптимізації вже знайшла свій шлях у багатьох областях моделювання та управління в техніці, і в наш час активно використовується в багатьох інших галузях прикладних наук, зокрема біології, медицині, економіці та фінансах. Дослідницька діяльність в оптимальному контролі розглядається як джерело багатьох корисних та гнучких інструментів, таких як оптимальна терапія (у медицині) та стратегії (в економіці). Методи теорії оптимального управління базуються на різноманітному спектрі математичних результатів, а, з іншого боку, проблеми управління забезпечують багате джерело глибоких математичних проблем. Вибір застосувань до наук про життя або економіки враховує сучасні тенденції лікування економічних проблем в осмосі за допомогою біологічних парадигм. Дана робота містить конкретний приклад моделювання реальних проблеми з біології, що ілюструє силу теорії управління в цій області. Математична біологія сягає своїм корінням в популяційну екологію, яка трактує математичне моделювання взаємодіючих видів за лініями, встановленими математиками А. Лоткою (1924) та В. Вольтеррою (1926), з точки зору нелінійних звичайних диференціальних рівнянь. Метою моделей «а-ля Лотка – Вольтерра» є кількісний опис еволюції взаємодіючих популяцій у часі. Однак є важливі аспекти, якими не можна нехтувати щодо просторової структури відповідних популяцій. Ця робота присвячений вивченню задачі оптимального управління динамічною моделлю популяції, пов'язаної з дифузійними моделями, тобто задача, яка регулюється диференціальними рівняннями в частинних похідних типу реакція-дифузія. Виведено умови існування оптимального керування та принцип максимуму, вказано числові алгоритми для наближення оптимальних значень функціоналів витрат. Представлена комп'ютерна програма, заснована на математичних результатах та побудована завдяки числовому наближенню методом градієнта.

2 Дифузія в математичних моделях

Пригадаємо два основних та універсальних методи введення дифузії. Перший - тривіальний наслідок із закону збереження об'єднаний з законом Fisk, інший пов'язаний з відповідним масштабуванням у часі та просторі випадкового процесу.

Розглянемо дифузію популяції чи іншої речовини, просторова щільність якої, в просторовому положенні x , в момент часу t , визначається як $y(x,t)$, (де $x \in \bar{\Omega} \subset \mathbb{R}^N$ - середовище існування, $N \in \mathbb{N}^*$ і $t \geq 0$). Функцію y , яка задає дифузію популяції в момент часу $t \geq 0$, в деякій області $V \subset \Omega$ (припустимо, що V -відкрита, обмежена підмножина і $V \in C^1$) можна подати таким чином

$$Y_V(t) = \int_V y(x,t) dx. \quad (1)$$

$$J(x,t) = (\gamma(x,t) \Delta_x y(x,t)) \cdot \nu(x)$$

де $\nu(x)$ -зовнішня нормаль до ∂V , $\gamma(x,t)$ -параметр дифузії і $\Delta_x y = y_x$ -градієнт y стосовно x .

Якщо ми позначимо через $f(x,t)$ деякі зміни в чисельності популяції $x \in V, t \geq 0$, тоді природний закон збереження для зміни чисельності популяції в середовищі V має вигляд

$$Y'_V(t) = \int_{\partial V} J(x,t) d\sigma + \int_V f(x,t) dx.$$

Застосовуючи теорему про диференцію і (1) отримаємо

$$\int_V \left[\frac{\partial y}{\partial t}(x,t) - \operatorname{div}_x(\gamma(x,t) \Delta_x y(x,t)) - f(x,t) \right] dx = 0.$$

Підобласть V - довільна, тому підінтегральна функція дорівнює нулю. Таким чином

$$\frac{\partial y}{\partial t}(x,t) - \operatorname{div}_x(\gamma(x,t) \Delta_x y(x,t)) - f(x,t) = 0. \quad (2)$$

У випадку, коли y - стала, рівняння (2) набуває вигляду

$$\frac{\partial y}{\partial t}(x,t) - \gamma \Delta y(x,t) = f(x,t), \quad (3)$$

де $\Delta = \Delta_x$ оператор Лапласа стосовно x . Рівняння Fisher - це частковий випадок (3), для окремої популяції і $f(x,t)$ описує процес народження і смерті:

$$\frac{\partial y}{\partial t}(x,t) - \gamma \Delta y(x,t) = -\mu y(x,t) + \beta y(x,t) \left(1 - \frac{y(x,t)}{k}\right) \quad (4)$$

Тут β - природний коефіцієнт народжуваності, μ - природний коефіцієнт смертності, а k - додатна стала. Це логістична модель, тому що - це додаткова смертність, як наслідок перенаселення, і вона пропорційна до щільності населення.

Тепер виведемо рівняння дифузії, як простого випадкового процесу. Простим випадковим процесом є ланцюг Маркова $(X_n)_{n \in \mathbb{N}}$ зі зліченим простором станів $E = \mathbb{Z}$, множина всіх цілих чисел і матриць переходу

$$\begin{aligned} \text{prob}(X_{n+1} = k+1 | X_n = k) &= p_{k,k+1} = p \\ \text{prob}(X_{n+1} = k-1 | X_n = k) &= p_{k,k-1} = 1-p \\ \text{prob}(X_{n+1} = j | X_n = k) &= p_{kj} = 0, \end{aligned}$$

для довільного $k \in E, j \in E, j \neq k-1, j \neq k+1, p \in (0, 1)$.

У випадку симетрії $p = \frac{1}{2}$.

Обравши за теоремою про граничну ймовірність, ланцюг Маркова набуде вигляду

$$\begin{aligned} \text{prob}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) \\ = \text{prob}(X_{n+1} = j | X_n = i) = p_{ij}, \end{aligned}$$

для довільних $i, j \in E$, ми отримаємо

$$p_k(n) = \frac{1}{2} p_{k+1}(n) + \frac{1}{2} p_{k-1}(n);$$

так, що

$$p_k(n+1) - p_k(n) = \frac{1}{2} [p_{k+1}(n) - 2p_k(n) + p_{k-1}(n)] \quad (5)$$

Якщо ми візьмемо $\Delta t \in \mathbb{R}_+^*$, — одиничний крок часу, $\Delta x \in \mathbb{R}_+^*$, — одиничний крок простору, тоді рівняння (5) можна переписати так

$$p_{k\Delta x}((n+1)\Delta t) - p_{k\Delta x}(n\Delta t) = \frac{1}{2}[p_{(k+1)\Delta x}(n\Delta t) - 2p_{k\Delta x}(n\Delta t) + p_{(k-1)\Delta x}(n\Delta t)]$$

Застосувавши наближення Taylor's другого порядку, остання рівність набуде вигляду

$$y(x, t + \Delta t) - y(x, t) = \frac{1}{2}(\Delta x)^2 \frac{\partial^2 y}{\partial x^2}(x, t) + o((\Delta x)^2),$$

з якого

$$\frac{y(x, t + \Delta t) - y(x, t)}{\Delta t} = \frac{1}{2} \frac{(\Delta x)^2}{\Delta t} \frac{\partial^2 y}{\partial x^2}(x, t) + \frac{o((\Delta x)^2)}{\Delta t}. \quad (6)$$

Нехай $\Delta t \rightarrow 0, \Delta x \rightarrow 0$, в такому випадку отримаємо, що

$$\frac{(\Delta x)^2}{\Delta t} = 2\gamma,$$

З (6) отримали звичайне рівняння дифузії із сталим коефіцієнтом дифузії

$$\frac{\partial y}{\partial t}(x, t) = \gamma \frac{\partial^2 y}{\partial x^2}(x, t), \quad x \in \mathbb{R}, t > 0.$$

Використавши початкову умову

$$y(\cdot, 0) = \delta_0$$

з теорії диференціальних рівнянь з частинними похідними, отримаємо фундаментальний розв'язок

$$y(x, t) = \frac{1}{\sqrt{(4\pi\gamma t)}} \exp\left(-\frac{1}{4\gamma t} x^2\right), \quad x \in \mathbb{R}, t \geq 0,$$

який є функцією щільності ймовірності Gaussian $N(0, 2\gamma t)$, що є розподілом поля Brownian руху в момент часу t .

Що стосується крайових умов, пов'язаних з рівняннями (3),(4) чи іншими нелінійними параболічними рівняннями, пригадаємо три, які найчастіше застосовують.

Умова Dirichlet має вигляд

$$y(x, t) = \phi(x, t), \quad x \in \partial\Omega, t \geq 0,$$

де ϕ – задана функція. Коли $\phi \equiv 0$, це визначає (для біологічної популяції) цілком неприйнятну межу $\partial\Omega$.

Умова Neumann має вигляд:

$$\frac{\partial y}{\partial \nu}(x, t) = \varphi(x, t), \quad x \in \partial\Omega, \quad t \geq 0,$$

де φ – задана функція. Для біологічної популяції ця умова означає, що потік популяції через межу φ . Якщо $\varphi \equiv 0$, тоді немає потоку популяції через межу і популяція є ізольованою в деякому середовищі Ω . Умова Robin має вигляд:

$$\frac{\partial y}{\partial \nu}(x, t) + \alpha(x, t) = \phi(x, t), \quad x \in \partial\Omega, \quad t \geq 0,$$

де α та ϕ – задані функції. Ця умова зазвичай впливає (для популяції) з того, що потік населення через межу пропорційний різниці між щільністю населення $y(x, t)$ всередині Ω та щільністю населення $\tilde{y}(x, t)$ за межами Ω :

$$\frac{\partial y}{\partial \nu}(x, t) = -\alpha(x, t)(y(x, t) - \tilde{y}(x, t)), \quad x \in \partial\Omega, \quad t \geq 0,$$

$$\frac{\partial y}{\partial \nu}(x, t) + \alpha(x, t)y(x, t) = \alpha(x, t)\tilde{y}(x, t), \quad x \in \partial\Omega, \quad t \geq 0.$$

3 Постановка задачі

Припустимо, що дві популяції можуть вільно взаємодіяти в одному середовищі проживання Ω . Наша проблема полягає в тому, що біомаса захопленої та з'їденої здобичі по x у середовищі проживання Ω надалі розподіляється по всьому ареалу, зайнятим видом хижаків, Ω . Це призводить до локальної (не локальної) міжвидової взаємодії двох видів на рівні хижака; тобто функціональна реакція на хижацтво є локальною, тоді як числова реакція на хижацтво нелокальна і розподілена по Ω ;

Виведемо математичну задачу, з якою ми працюємо. Нехай $y_1(x, t)$ - щільність у положенні x та час t видобутого виду, розподіленого по просторовій області $\Omega \subset \mathbb{R}^N, N = 1, 2$ або 3 , і припустимо, що його просторово-часова динаміка регулюється рівнянням

$$\frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = r_1 y_1, \quad x \in \Omega, \quad t \in (0, T),$$

де $r_1 > 0$ - природна швидкість приросту, $d_1 > 0$ - коефіцієнт дифузії, і $T > 0$. Нехай $y_2(x, t)$ - щільність у положенні x та час t виду хижаків, розподілених на одному ареалі; за відсутності вищезгаданої здобичі - яка вважається її унікальним ресурсом - популяція хижаків буде занепадати зі експоненційною швидкістю $r_2 > 0$, а її просторово-часова динаміка регулюватиметься базовою лінійною моделлю,

$$\frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = r_2 y_2, \quad x \in \Omega, \quad t \in (0, T),$$

де $d_2 > 0$ - коефіцієнт дифузії. За наявності обох популяцій хижацтво відбувається на Ω . Динаміка здобичі змінюється хижацтвом

$$\frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = r_1 y_1 - \mu u(x, t) y_1 y_2, \quad x \in \Omega, \quad t \in (0, T), \quad (7)$$

де $\mu_1 > 0$ і $1 - u(x, t)$ - сегрегація двох сукупностей, u - оптимальний є контроль.

Позначимо $y_1(t)$ щільність популяції здобичі в момент t та $y_2(t)$ щільність хижаків у момент часу t , де (y_1^u, y_2^u) є розв'язком системи Лотка - Вольтерра ($T > 0$), що описує динаміку системи хижак - жертва на інтервалі часу $(0, T)$.

$$\begin{cases} \frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = r_1 y_1 - \mu_1 u(x, t) y_1 y_2, (x, t) \in \Omega \times \in (0, T) \\ \frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = -r_2 y_2 + \mu_2 u(x, t) y_1 y_2, (x, t) \in \Omega \times \in (0, T) \end{cases} \quad (8)$$

(y_1^u, y_2^u) це рішення наступної задачі з початковим значенням на $(L^2(\Omega) \times L^2(\Omega))$.

$r_1 > 0$ - це темп зростання здобичі за відсутності хижаків.

$r_2 > 0$ - це швидкість помирання популяції хижаків за відсутності здобичі.

μ_1 і μ_2 - додатні константи; d_1 та d_2 - коефіцієнти дифузії; $1 - u(x, t)$

- відокремлення двох популяцій, u - оптимальний контроль; $\mu_1 y_1(t)$ - додатковий рівень смертності жертви в момент t , внаслідок хижацтва (це пропорційно густоті населення хижака); $i \mu_2 y_2(t)$ - додатковий темп зростання здобичі в момент t , обумовлений наявністю здобичі (вона пропорційна до густоти населення здобичі).

Видобута і з'їдена здобич за час $t > 0$ та місцеположення $x' \in \Omega$ перетворюються на біомасу за допомогою коефіцієнта перетворення, даючи чисельну відповідь на хижацтво

$$\mu_2 u(x', t) y_1(x', t) y_2(x', t) (\mu_2 > 0).$$

Будемо вважати, що ця кількість розподіляється на Ω через загальне невід'ємне ядро

$$l(x, x') (l \in L^\infty(\Omega \times \Omega), l(x, x') \geq 0 \quad (x, x') \in (\Omega \times \Omega))$$

тож $l(x, x') \mu_2 u(x', t) y_1(x', t) y_2(x', t)$ - це біомаса, розподілена при $x \in \Omega$, що виникає внаслідок хижацтва при $x' \in \Omega$. Збереження біомаси передбачає, що умови консистенції повинні виконуватися:

$$\int_{\Omega} l(x, x') dx = 1, \quad x' \in \Omega$$

Далі зчитуємо динаміку хижака

$$\frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = -r_2 y_2 + \mu_2 \int_{\Omega} l(x, x') u(x', t) y_2(x', t) dx', \quad x \in \Omega, t \in (0, T) \quad (9)$$

Для завершення нашої моделі повинні бути встановлені граничні умови для обох видів. Ми вибираємо граничні умови, що відповідають ізольованим популяціям:

$$\frac{\partial y_1}{\partial \nu}(x, t) = \frac{\partial y_2}{\partial \nu}(x, t) = 0, \quad x \in \Omega, \quad t \in (0, T). \quad (10)$$

Далі негативні та обмежені початкові умови призначаються в часі $t = 0$:

$$\begin{cases} y_1(x, 0) = y_{01}(x), & x \in \Omega \\ y_2(x, 0) = y_{02}(x), & x \in \Omega \end{cases} \quad (11)$$

Тоді (8) - (11) є базовою моделлю для нашої системи хижак-здобич.

Припустимо, що

$$y_{01}, y_{02} \in L^\infty(\Omega),$$

та

$$y_{01}(x) > 0, y_{02}(x) > 0 \quad x \in \Omega.$$

Початкові умови:

$$\begin{cases} y'(t) = f(t, u(t), y(t)), t \in (0, T) \\ y(0) = y_0, \end{cases} \quad (12)$$

де

$$y_0 = \begin{pmatrix} y_{01} \\ y_{02} \end{pmatrix},$$

$$f(t, u, y) = Ay + \begin{pmatrix} r_1 y_1 - \mu_1 u y_1 y_2 \\ -r_2 y_2 + \mu_2 u y_1 y_2 \end{pmatrix}.$$

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

$$D(A) = \{\omega = (\omega_1, \omega_2) \in H^2(\Omega) \times H^2(\Omega) : \frac{\partial \omega_1}{\partial \nu} = \frac{\partial \omega_2}{\partial \nu} = 0, \text{ на } \partial\Omega\},$$

$$Ay = \begin{pmatrix} d_1 \Delta y_1 \\ d_2 \Delta y_2 \end{pmatrix}, y \in D(A).$$

Ми зацікавлені в максимізації загальної кількості особин обох популяцій в момент $T > 0$. Проблема може бути переформульована як

$$\max \Psi(u) = \int_{\Omega} y_1''(x, T) + y_2''(x, T) dx, \quad (13)$$

$$u \in L^2(\Omega \times (0, T)), 0 \leq u(x, t) \leq 1, t \in (0, T),$$

де (y_1'', y_2'') є розв'язком (12).

Теорема 1 Якщо $(u^*, (y_1^*, y_2^*))$ оптимальна пара для (13), і якщо $p =$

(p_1, p_2) - рішення для

$$\left\{ \begin{array}{l} \frac{\partial p_1}{\partial t} + d_1 \Delta p_1 = -r_1 p_1 + \mu_1 u^* y_2^* p_1 - \mu_2 u^*(x, t) y_2^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx', \\ \frac{\partial p_2}{\partial t} + d_2 \Delta p_2 = r_2 p_2 + \mu_1 u^* y_1^* p_1 - \mu_2 u^*(x, t) y_1^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx' \\ \frac{\partial p_1}{\partial \nu}(x, T) = \frac{\partial p_2}{\partial \nu}(x, T) = 0, \quad (x, t) \in \partial \Omega \times (0, T) \\ p_1(x, T) = p_2(x, T) = 1, \quad x \in \Omega \end{array} \right.$$

$(x, t) \in \Omega \times (0, T)$

Тоді

$$u^*(x, t) = \begin{cases} 0, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) < 0, \quad (x, t) \in \Omega \times (0, T) \\ 1, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) > 0, \quad (x, t) \in \Omega \times (0, T) \end{cases}$$

Ця теорема забезпечує необхідні умови оптимальності першого порядку.

Доведення подане нище.

4 Існування оптимального контролю

Визначимо

$$\max \Psi(u) = \int_{\Omega} y_1^u(x, T) + y_2^u(x, T) dx \quad (14)$$

Нехай

$$d = \sup_{u \in \Omega} \Phi(u) \quad (15)$$

Очевидно, що $d \in \mathbb{R}^+$. Для будь-яких $n \in \mathbb{N}^*$, існує $u_n \in \Omega$ таке що

$$d - \frac{1}{n} < \Phi(u_n) \leq d \quad (16)$$

Використовуючи принцип порівняння для параболічних рівнянь, ми отримаємо наступне

$$0 < y_1^u(x, t) \leq y_1^0(x, t) \quad (x, t) \in \Omega \times (0, T),$$

$$0 < y_2^u(x, t) \leq y_2^0(x, t) \quad (x, t) \in \Omega \times (0, T).$$

Отже,

$$0 \leq \int_{\Omega} y_1^u(x, T) + y_2^u(x, T) dx \leq L \int_{\Omega} y_1^0(x, T) + y_2^0(x, T) dx$$

Оскільки, $\{u_n\}_{n \in \mathbb{N}^*}$ є обмеженою послідовністю в $L^2(\omega \times (0, T))$, з цього випливає, що існує підпослідовність, також позначена як $\{u_n\}_{n \in \mathbb{N}^*}$

$$u_n \rightarrow u^* \text{ слабко збіжне на } L^2(\omega \times (0, T)),$$

та

$$tu_n \rightarrow tu^* \text{ слабко збіжне на } L^2(\omega \times (0, T)).$$

За теоремою Арцели та враховуючи, що $\{u_n\}_n$ обмежена в $L^2(0, T)$, ми отримуємо, що

$$\begin{aligned} u_{n_k} &\rightarrow u^* \text{ слабко на } L^2(\omega \times (0, T)) \\ y_1^{u_{n_k}} &\rightarrow y_1^* \quad C([0, T]) \\ y_2^{u_{n_k}} &\rightarrow y_2^* \quad C([0, T]) \end{aligned} \quad (17)$$

($u^* \in \Omega$, тому що Ω - замкнута опукла підмножина на $L^2(\omega \times (0, T))$, а отже, слабко збіжна).

Якщо ми позначимо

$$a_n(x, t) = r_1 y_1^{u_n}(x, t) - \mu_1 u_n(x, t) y_1^{u_n}(x, t) y_2^{u_n}(x, t),$$

$$b_n(x, t) = -r_2 y_2^{u_n}(x, t) + \mu_2 u_n(x, t) y_1^{u_n}(x, t) y_2^{u_n}(x, t),$$

тоді очевидно, що $\{a_n\}_{n \in \mathbb{N}^*}$, $\{b_n\}_{n \in \mathbb{N}^*}$ є обмежені на $L^\infty(Q_T)$, (і на $L^2(Q_T)$)

також) та $(y_1^{u_n}, y_2^{u_n})$ є розв'язком (18)

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = a_n(x, t), \quad (x, t) \in Q_T \\ \frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = b_n(x, t) \quad (x, t) \in Q_T \\ \frac{\partial y_1}{\partial \nu}(x, T) = \frac{\partial y_2}{\partial \nu}(x, T) = 0, \quad x \in \Sigma_T \\ y_1(x, 0) = y_{01}(x), y_2(x, 0) = y_{02}(x), \quad x \in \Omega. \end{array} \right. \quad (18)$$

Обмеженість $\{a_n\}_{n \in \mathbb{N}^*}$, $\{b_n\}_{n \in \mathbb{N}^*}$ передбачає (через результат компактності для параболічних рівнянь), що існує підпоследовність, така що

$$\left\{ \begin{array}{l} a_{n_k} \rightarrow a^*, \quad L^2(Q_T) \\ b_{n_k} \rightarrow b^*, \quad L^2(Q_T) \\ y_1^{u_{n_k}} \rightarrow y_1^*, \quad Q_T \\ y_2^{u_{n_k}} \rightarrow y_2^*, \quad Q_T \end{array} \right. \quad (19)$$

тож (y_1^*, y_2^*) є розв'язком наступної системи

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = a^*(x, t), \quad (x, t) \in Q_T \\ \frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = b^*(x, t) \quad (x, t) \in Q_T \\ \frac{\partial y_1}{\partial \nu}(x, T) = \frac{\partial y_2}{\partial \nu}(x, T) = 0, \quad x \in \Sigma_T \\ y_1(x, 0) = y_{01}(x), y_2(x, 0) = y_{02}(x), \quad x \in \Omega. \end{array} \right. \quad (20)$$

З іншого боку, за (19) та за допомогою слабкої збіжності $\{u_n\}_n \in \mathbb{N}^*$ на $L^\infty(Q_T)$ та обмеженості $\{y^{u_n}\}_n \in \mathbb{N}^*$ в $L^\infty(Q_T)$, ми можемо зробити висновок про те, що

$$\begin{aligned} a_{n_k} &\rightarrow r_1 y_1^* - \mu_1 u^* y_1^* y_2^* \text{ на } L^2(Q_T), \\ b_{n_k} &\rightarrow -r_2 y_2^* + \mu_2 u^* y_1^* y_2^* \text{ на } L^2(Q_T), \end{aligned}$$

і як наслідок

$$a_{n_k} = r_1 y_1^* - \mu_1 u^* y_1^* y_2^* \text{ на } L^2(Q_T),$$

$$b_{n_k} = -r_2 y_2^* + \mu_2 u^* y_1^* y_2^* \text{ на } L^2(Q_T).$$

На закінчення $y^* = (y_1^*, y_2^*)$ є рішенням (12), що відповідає $u := u^*$; то, $y_1^* = y_1^{u^*}, y_2^* = y_2^{u^*}$. Якщо ми переходимо до границі в (16), ми отримуємо

$$d = \Phi(u^*),$$

і що u^* є оптимальним управлінням для задачі (13).

5 Принцип максимуму

Нехай маємо наступну систему рівнянь

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial t} - d_1 \Delta y_1 = r_1 y_1 - \mu_1 u(x, t) y_1 y_2, \quad (x, t) \in \Omega \times (0, T) \\ \frac{\partial y_2}{\partial t} - d_2 \Delta y_2 = -r_2 y_2 + \mu_2 u(x, t) y_1 y_2, \quad (x, t) \in \Omega \times (0, T) \\ \frac{\partial y_1}{\partial \nu}(x, t) = \frac{\partial y_2}{\partial \nu}(x, t) = 0, \quad x \in \Omega, \quad t \in (0, T). \\ y_1(x, 0) = y_{01}(x), y_2(x, 0) = y_{02}(x), \quad x \in \Omega \end{array} \right. \quad (21)$$

Систему (21) можна переписати у вигляді початкової задачі в $L^2(\Omega)$.

$$\left\{ \begin{array}{l} y'(t) = f(t, u(t), y(t)), \quad t \in (0, T) \\ y(0) = y_0, \end{array} \right.$$

де

$$y_0 = \begin{pmatrix} y_{01} \\ y_{02} \end{pmatrix},$$

$$f(t, u, y) = Ay + \begin{pmatrix} r_1 y_1 - \mu_1 u y_1 y_2 \\ -r_2 y_2 + \mu_2 u y_1 y_2 \end{pmatrix}.$$

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

Тут A - лінійний необмежений оператор. Насправді A визначається як $D(A) = \{\omega = (\omega_1, \omega_2) \in H^2(\Omega) \times H^2(\Omega) : \frac{\partial \omega_1}{\partial \nu} = \frac{\partial \omega_2}{\partial \nu} = 0, \text{ на } \partial\Omega\}$,

$$Ay = \begin{pmatrix} d_1 \Delta y_1 \\ d_2 \Delta y_2 \end{pmatrix}, y \in D(A).$$

Також зауважте, що A є самоспряженим оператором, і тому A^* може бути замінено на A . Власне, ось результат, який ми доводимо.

Теорема 2 Якщо $(u^*, (y_1^*, y_2^*))$ оптимальна пара для (13), і якщо $p = (p_1, p_2)$ - рішення для

$$\begin{cases} \frac{\partial p_1}{\partial t} + d_1 \Delta p_1 = -r_1 p_1 + \mu_1 u^* y_2^* p_1 - \mu_2 u^*(x, t) y_2^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx', \\ \frac{\partial p_2}{\partial t} + d_2 \Delta p_2 = r_2 p_2 + \mu_1 u^* y_1^* p_1 - \mu_2 u^*(x, t) y_1^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx' \\ \frac{\partial p_1}{\partial \nu}(x, T) = \frac{\partial p_2}{\partial \nu}(x, T) = 0, \quad (x, t) \in \partial \Omega \times (0, T) \\ p_1(x, T) = p_2(x, T) = 1, \quad x \in \Omega \end{cases}$$

Тоді

$$u^*(x, t) = \begin{cases} 0, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) < 0 \\ 1, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) > 0, \end{cases}$$

Доведення:

Розглянемо множину

$V = \{\omega \in L^2(\omega \times (0, T)); \quad u^* + \varepsilon \omega \in K \text{ для будь-яких } \varepsilon > 0 \text{ достатньо малого}\}$. Нехай (z_1, z_2) розв'язок наступної системи рівнянь Для довільного, але фіксованого $v \in V$ маємо

$$\begin{cases} p_1' = r_1 z_1 - \mu_1 u^* y_1^* z_1 - \mu_1 u^* y_2^* z_2 - \mu_1 v y_1^* y_2^* \int_{\Omega} l(x', x) z_2(x', t) dx', & t \in (0, T) \\ z_2' = -r_2 z_2 + \mu_2 u^* y_1^* z_1 + \mu_2 u^* y_2^* z_2 + \mu_2 v y_1^* y_2^* \int_{\Omega} l(x', x) z_2(x', t) dx', & t \in (0, T) \\ z_1(0) = z_2(0) = 0, \quad x \in \Omega. \end{cases} \quad (22)$$

Оскільки

$$\int_{\omega} y_1^*(T) + y_2^*(T) \geq \int_{\omega} y_1^{u^* + \varepsilon v}(T) + y_2^{u^* + \varepsilon v}(T),$$

То ми отримаємо

$$\int \frac{y_1^{u^*+\varepsilon v}(T) - y_1^*}{\varepsilon} + \int \frac{y_2^{u^*+\varepsilon v}(T) - y_2^*}{\varepsilon} \leq 0, \quad (23)$$

для будь-якого $\varepsilon > 0$ досить малого.

Для $\varepsilon > 0$ досить малого ми маємо, що $y^{u^*+\varepsilon v}$ задовольняє

$$(y^{u^*+\varepsilon v})'(t) \leq r_1 y^{u^*+\varepsilon v}(t), \quad t \in (0, T),$$

і, отже, впливає, що існує $M \in (0, +\infty)$ така, що

$$0 \leq y^{u^*+\varepsilon v}(t) \leq M, t \in [0, T]$$

для будь-якого $\varepsilon > 0$ досить малого. З іншого боку

$$(y^{u^*+\varepsilon v})'(t) \leq (-r_2 + M\mu_2)y^{u^*+\varepsilon v}(t), t \in (0, T)$$

і це означає, що $y^{u^*+\varepsilon v}(t)$ обмежена в $C([0, T])$ (при $\varepsilon > 0$ досить малому). Звідси впливає, що обидві послідовності $y_1^{u^*+\varepsilon v}(t)$ та $y_2^{u^*+\varepsilon v}(t)$ рівномірно обмежені та рівномірно неперервні на $[0, T]$.

Лема 1 Справедливі наступні збіжності:

$$\begin{aligned} y^{u^*+\varepsilon v}(T) &\rightarrow y^{u^*} \text{ на } L^\infty(Q_T) \\ \frac{1}{\varepsilon} [y^{u^*+\varepsilon v}(T) - y^{u^*}] &\rightarrow z \text{ на } L^\infty(Q_T) \end{aligned}$$

$\varepsilon \rightarrow 0+$. Якщо ми перейдемо до границі в (23) (і використаємо Теорему Лебега та лему (1)), то ми зможемо зробити висновок про те,що

$$\int_{\Omega} z_1(T) + z_2(T) dx \leq 0$$

Нехай $p = (p_1, p_2)$ є розв'язком наступної системи рівнянь

$$\left\{ \begin{array}{l} \frac{\partial p_1}{\partial t} + d_1 \Delta p_1 = -r_1 p_1 + \mu_1 u^* y_2^* p_1 - \mu_2 u^*(x, t) y_2^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx', \\ \frac{\partial p_2}{\partial t} + d_2 \Delta p_2 = r_2 p_2 + \mu_1 u^* y_1^* p_1 - \mu_2 u^*(x, t) y_1^*(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx' \\ \frac{\partial p_1}{\partial \nu}(x, T) = \frac{\partial p_2}{\partial \nu}(x, T) = 0, \quad (x, t) \in \partial \Omega \times (0, T) \\ p_1(x, T) = p_2(x, T) = 1, \quad x \in \Omega \end{array} \right. \quad (24)$$

Помноживши перше рівняння із системи на z_1 та друге на z_2 та інтегруючи на $[0, T]$ ми отримаємо, що

$$\begin{aligned} & \int_{\Omega} \int_0^T [p_1'(x, t) z_1(x, t) + p_2'(x, t) z_2(x, t)] dt = \\ & \int_{\Omega} \int_0^T [-r_1 p_1(x, t) z_1(x, t) + \mu_1 u^*(x, t) y_2^*(x, t) p_1(x, t) z_1(x, t) - \mu_2 u^*(x, t) p_2(x, t) z_1(x, t) \\ & + \mu_1 u^*(x, t) y_1^*(x, t) p_1(x, t) z_2(x, t) - \mu_2 u^*(x, t) y_2^*(x, t) p_2(x, t) z_2(x, t) + r_2 p_2(x, t) z_2(x, t)] dt. \\ & \int_{\Omega} [p_1(x, T) z_1(x, T) + p_2(x, T) z_2(x, T) - p_1(x, 0) z_1(x, 0) - p_2(x, 0) z_2(x, 0)] dt = \\ & = \int_{\Omega} \int_0^T y_1^*(x, t) y_2^*(x, t) v(x, t) [\mu_2 p_2(x, t) - \mu_1 p_1(x, t)] dt \leq 0, \end{aligned}$$

Як наслідок, ми отримуємо що

$$\int_{\Omega} z_1(T) + z_2(T) dx = \int_{\Omega} \int_0^T y_1^*(x, t) y_2^*(x, t) v(x, t) [\mu_2 p_2(x, t) - \mu_1 p_1(x, t)] dx dt \leq 0,$$

для будь-якого $v \in V$, що означає

$$u^*(x, t) = \begin{cases} 0, & y_1^*(x, t) y_2^*(x, t) \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) < 0 \\ 1, & y_1^*(x, t) y_2^*(x, t) \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) > 0, \end{cases}$$

Оскільки y_1^*, y_2^* додатні функції, ми можемо зробити висновок

$$u^*(x, t) = \begin{cases} 0, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) < 0 \\ 1, & \mu_2 \int_{\Omega} l(x', x) p_2(x', t) dx' - \mu_1 p_1(x, t) > 0, \end{cases} \quad (25)$$

За (24) та (25) ми отримуємо в цьому випадку, що p є єдиним рішенням до Якщо ми апроксимуємо рішення p цієї задачі (яке не залежить від u^* або від y^{u^*} , то використовуючи (25), ми отримуємо u^* . Ось алгоритм.

- ОР1: Обчислити p як розв'язок задачі (24).
- ОР2: Обчислити оптимальний контроль за формулою (25).

Звичайно, обчислюваний контроль все ще є наближенням через те, що Проблема (ОР1) обчислюється чисельно. Але алгоритм досить простий. З (23) та (24) ми також можемо зробити висновок, що u - це контроль, при якому оптимальне управління u^* явно не залежить від y_0 .

6 Градієнтний метод

Перед описом методу, слід спочатку описати задачу. Маємо непорожню, опуклу та замкнену множину $K \subset V$, f -диференційована і опукла функція, розглядаємо проблему мінімізації функції $f: V \rightarrow \mathbb{R}$ на множині K : Знайти $u \in K$ таке, що

$$f(u) \leq f(v) \text{ для будь якого } v \in K \quad (26)$$

Проблема може бути записана еквівалентно як $\text{Inf}\{f(u); u \in K\}$. Існування існування розв'язку подано в Теоремі:

1. Якщо f -опукла, K -опукла, непорожня та замкнута то задача мінімізації функції f має розв'язок.
2. Якщо f строго опукла то рішення цієї задачі єдине.

Розглянемо ще одну, не менш важливу Теорему:
Нехай f слабо диференційована:

1. Якщо $u^* \in K$ є оптимальним, тоді для будь якого $\lambda > 0$

$$P_K(u^* - \lambda \nabla f(u^*)) = u^*. \quad (27)$$

2. Якщо f також опукла і якщо тут існує $\lambda > 0$ для якого виконується умова з першого пункту, тоді u^* є також розв'язком задачі мінімізації функції f .

Теорема 3 $\omega \in V$ є напрямком спадання f при $u \in V$ якщо тут існує $\bar{\rho} > 0$ таке що:

$$f(u + \rho \omega) < f(u) \text{ для будь якого } \rho \in (0, \bar{\rho}]. \quad (28)$$

Напрямок спадання характеризується наступною Теоремою: Припустимо що f має непервний градієнт на V . Якщо для $\omega \in V$ виконується

$$(\nabla f(u), \omega) < 0 \quad (29)$$

тоді ω є напрямком спадання функції f при u .

Отже ідея методів прогнозування градієнта така: задаємо напрямком спадання ω_k , тобто напрямком який задовільняє умови

$$(\nabla f(u_k), \omega_k) < 0, \quad (30)$$

шукаємо крок ρ_k таким чином, щоб значення вектора $u_k + \rho \omega_k$ було мінімальним, тобто

$$f(P_K(u_k + \rho_k \omega_k)) \leq f(P_K(u_k + \rho \omega_k)) \text{ для будь якого } \rho > 0. \quad (31)$$

7 Алгоритм градієнта для проблеми (*)

Нижче подано прогнозований алгоритм градієнтного типу:

S0: Вибираємо $u^0 \in K; j := 0;$

S1: Обчислюємо $y^{(j)} = y_1^{(j)}, y_2^{(j)}$ розв'язок 1-3 із введенням $u^{(j)}$:

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial t} + d_1 \Delta y_1 = -r_1 y_1 + \mu_1 u^{(j)}(x, t) y_1 y_2, \quad x \in \Omega, t \in (0, T) \\ \frac{\partial y_2}{\partial t} + d_2 \Delta y_2 = -r_2 y_2 + \mu_2 \int_{\Omega} l(x', x) u^{(j)}(x', t) y_1(x', t) y_2(x', t) dx', \\ \frac{\partial y_1}{\partial \nu}(x, T) = \frac{\partial y_2}{\partial \nu}(x, T) = 0, \quad x \in \partial \Omega, t \in (0, T) \\ y_1(x, 0) = y_{01}(x), y_2(x, 0) = y_{02}(x), \quad x \in \Omega. \end{array} \right. \quad (32)$$

S2: Обчислюємо $p^{(j)} = (p_1^{(j)}, p_2^{(j)})$, що є розв'язком наступної системи рівнянь:

$$\left\{ \begin{array}{l} \frac{\partial p_1}{\partial t} + d_1 \Delta p_1 = -r_1 p_1 + \mu_1 u^{(j)} y_2^{(j)} p_1 - \mu_2 u^{(j)}(x, t) y_2^{(j)}(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx', \quad x \in \Omega, t \in (0, T) \\ \frac{\partial p_2}{\partial t} + d_2 \Delta p_2 = r_2 p_2 + \mu_1 u^{(j)} y_1^{(j)} p_1 - \mu_2 u^{(j)}(x, t) y_1^{(j)}(x, t) \int_{\Omega} l(x', x) p_2(x', t) dx', \quad x \in \partial \Omega, t \in (0, T) \\ \frac{\partial p_1}{\partial \nu}(x, T) = \frac{\partial p_2}{\partial \nu}(x, T) = 0, \quad x \in \partial \Omega, t \in (0, T) \\ p_1(x, T) = p_2(x, T) = 1, \quad x \in \Omega. \end{array} \right.$$

S3: Обчислюємо $v^{(j)}$ відповідно до Теорема 1:

$$v^{(j)}(x, t) = \left\{ \begin{array}{l} 0, \mu_2 \int_{\Omega} l(x', x) p_2^{(j)}(x', t) dx' - \mu_1 p_1^{(j)}(x, t) < 0 \\ 1, \mu_2 \int_{\Omega} l(x', x) p_2^{(j)}(x', t) dx' - \mu_1 p_1^{(j)}(x, t) \geq 0, \quad \Omega \times (0, T). \end{array} \right.$$

S4: Обчислюємо $\lambda_j \times [0, 1]$, що є розв'язком проблеми

$$\max \Psi(\lambda u^{(j)} + (1 - \lambda) v^{(j)}), \lambda \in [0, 1]$$

S5: Обчислюємо $u^{(j+1)}$ як

$$u^{(j+1)} = \lambda_j u^{(j)} + (1 - \lambda_j) v^{(j)}.$$

S6: Припиняємо обчислення, коли

$$\|u^{(j+1)} - u^{(j)}\| < \varepsilon,$$

інакше $j := j + 1$; переходимо до S1.

Ми нагадаємо, що опукла комбінація двох елементів керування, які приймають значення 0 та 1, не обов'язково приймають лише ці два значення. Ось чому краще замінити опуклу комбінацію $\lambda u^{(j)} + (1 - \lambda) v^{(j)}$ на S4 і S5 опуклою комбінацією точок $u^{(j)}$ і $v^{(j)}$.

8 Реалізація задачі у пакеті Python

Нехай $(x_1, x_2, t) \in \Omega \times (0, T)$; $(x_1, x_2) \in \Omega = (0, L_1) \times (0, L_2)$;

$$r_2 = 0,6 \quad r_1 = 0,07 \quad \mu_1 = 1; \quad d_1 = d_2 = -1; \quad \mu_2 = 2;$$

$$x = (x_1, x_2), \quad x' = (x'_1, x'_2) \quad L_1 = L_2 = 1; \quad T = 0,2;$$

$$l(x, x') = \left(\frac{\pi}{2} - x_1\right) \cos\left(\left(\frac{\pi}{2} - x_1\right)x'_1 + x_1\right) + \sin x_1 + x_2 \sin(x_2(1 - x'_2)) + \cos x_2;$$

$$N_{x_1} = N_{x_2} = 12; \quad N_t = 99; \quad y_{01} = 1, y_{02} = 2;$$

```

8 from scipy import integrate
9 import numpy as np
10 from math import *
11 import matplotlib.pyplot as plt
12 from mpl_toolkits.mplot3d import Axes3D
13 import scipy.optimize as optimize
14 from scipy import sparse
15 from matplotlib import cm
16
17 #1st equation y = (y1,y2), x = (x1,x2)
18 #N = 2, x = (x1,x2)
19 Lx1 = Lx2 = [0,1]
20 Nx1 = Nx2 = 12
21 Nt = 99
22 T = .3 #>0
23 d1 = 1 #>0, diffusion coef
24 d2 = 1 #>0
25 nu1 = 1 #>0
26 nu2 = 2 #>0
27 r1 = .07 #natural growth rate, >0
28 r2 = .6
29 eps = .01 #||u[k+1] - u[k]||<eps then u[k+1] optimal control
30
31
32
33 def u(x1,x2,t): return x1*x2 # optimal control 0 <= u(x1,x2) <= 1
34 def y1_init(x1,x2): return 1 #sin(x1-x2) + cos(x1+x2)#sin(x1+x2) #boundary condition
35 def y2_init(x1,x2): return 1 #sin(x1-x2)
36
37 #2nd system p=(p1,p2)
38 def free_y(y1,y2,u_j,r1=r1,r2=r2,nu1=nu1,nu2=nu2,Nx = Nx1,Ny=Nx2):
39     """
40     r2=nu2 = 0 => equation of 1st f from S1
41     r1=nu1 = 0 => equation of 2nd f from S1
42     """
43     #u = np.zeros((Nx+1, Ny+1))
44     # u = r1*np.array(y1) - r2*np.array(y2) - nu1*u_j*np.multiply(y1,y2) + nu2*u_j
45     u = r1*np.array(y1) - r2*np.array(y2) - nu1*u_j*y1*y2 + nu2*u_j*y1*y2
46     return u

```

```

49 def CondX1(y): # Coefficients for left and right boundaries
50     alf_min = 0e0; bet_min = 1e0; gam_min = 0e0
51     alf_max = 0e0; bet_max = 1e0; gam_max = 0e0
52     return (alf_min, bet_min, gam_min, alf_max, bet_max, gam_max)
53
54 def gold_bound(f,a,b,h=0.1):
55     delta = h
56     r=2.0/(3+sqrt(5)) #r-золоте відношення
57     c=a+r*(b-a)
58     d=b-r*(b-a)
59     l=[a,b,c,d]
60     arr_p= np.array(l)
61     f_arr_p=f(arr_p)
62     lf_arr_p=list(f_arr_p)
63     inx=lf_arr_p.index(min(lf_arr_p))
64     dpm=abs(arr_p-l[inx])
65     l_dpm=list(dpm)
66     indx=l_dpm.index(max(l_dpm))
67     arr_p3=np.delete(l,indx)
68     arr_p3=np.sort(arr_p3)
69     dv=abs(b-a)
70     a=arr_p3[0]
71     b=arr_p3[2]
72     c=arr_p3[1]
73     d=a+b-c
74     l=[a,b,c,d]
75     step=0
76     while dv>delta:
77         arr_p=np.array(l)
78         f_arr_p=f(arr_p)
79         lf_arr_p=list(f_arr_p)
80         inx=lf_arr_p.index(min(lf_arr_p))
81         dpm=abs(arr_p-l[inx])
82         l_dpm=list(dpm)
83         indx=l_dpm.index(max(l_dpm))
84         arr_p3=np.delete(l,indx)
85         arr_p3=np.sort(arr_p3)
86         dv=abs(b-a)
87         a=arr_p3[0]
88         b=arr_p3[2]

```

```

86     dv=abs(b-a)
87     a=arr_p3[0]
88     b=arr_p3[2]
89     c=arr_p3[1]
90     d=a+b-c
91     l=[a,b,c,d]
92     step+=1
93     res = np.array([c,f(c)])
94     return res[0]
95 #-----
96 def CondX2(x): # Coefficients for lower and upper boundaries
97     alf_min = 0e0; bet_min = 1e0; gam_min = 0e0
98     alf_max = 0e0; bet_max = 1e0; gam_max = 0e0
99     return (alf_min, bet_min, gam_min, alf_max, bet_max, gam_max)
100
101 def solver_dense(In_cond,a, f,backward = 0, Lx=Lx1, Ly=Lx2, Nx=Nx1, Ny=Nx2, Nt=Nt,
102 """
103 Solve  $u_t = a*(u_{xx} + u_{yy}) + f$ ,  $u(x,y,0)=I(x,y)$ , with boundary condition
104 from function CondX1 and CondX2
105 on the boundary, on  $Lx \times Ly \times [0,T]$ , with time step dt,
106 using the theta-scheme.
107 """
108 x = np.linspace(Lx[0], Lx[-1], Nx+1) # mesh points in x dir
109 y = np.linspace(Ly[0], Ly[-1], Ny+1) # mesh points in y dir
110 dx = x[1] - x[0] #dx = L/Nx
111 dy = y[1] - y[0] #dy = L/Ny
112 dt = T/Nt
113 t = np.linspace(0+backward*T, T-backward*T, Nt+1) # mesh points in time
114 # Mesh Fourier numbers in each direction
115 Fx = a*dt/dx**2
116 Fy = a*dt/dy**2
117 r = dt/(dx**2)
118 s = 1-2*r
119 if r>1/2: print('Input parameters cause instability!')
120 #Initialization
121 u = np.zeros((Nx+1, Ny+1)) # unknown u at new time level
122 u_n = np.zeros((Nx+1, Ny+1)) # u at the previous time level
123 #indexes

```



```

124 Ix = range(0, Nx+1)
125 Iy = range(0, Ny+1)
126 It = range(0, Nt+1)
127 # Load initial condition into u_n
128 for i in Ix:
129     for j in Iy:
130         u_n[i,j] = In_cond(x[i], y[j])
131 #Init boundaries
132 hx = dx # 2*Lx/(Nx-1)
133 hy = dy #2*Ly/(Ny-1)
134 CondY = CondX2
135 CondX = CondX1
136 betXmin = [0]*(Ny+1); betXmax = [0]*(Ny+1)
137 gamXmin = [0]*(Ny+1); gamXmax = [0]*(Ny+1)
138 betYmin = [0]*(Nx+1); betYmax = [0]*(Nx+1)
139 gamYmin = [0]*(Nx+1); gamYmax = [0]*(Nx+1)
140
141 # boundary conditions
142 for i in Ix: # lower and upper boundaries
143     (alf_min,bet_min,gam_min,alf_max,bet_max,gam_max) = CondY(x[i])
144     betYmin[i] = bet_min/(alf_min*hy + bet_min)
145     gamYmin[i] = gam_min/(alf_min + bet_min/hy)
146     betYmax[i] = bet_max/(alf_max*hy + bet_max)
147     gamYmax[i] = gam_max/(alf_max + bet_max/hy)
148
149 for j in Iy[1:-1]: # left and right boundaries
150     (alf_min,bet_min,gam_min,alf_max,bet_max,gam_max) = CondX(y[j])
151     betXmin[j] = bet_min/(alf_min*hx + bet_min)
152     gamXmin[j] = gam_min/(alf_min + bet_min/hx)
153     betXmax[j] = bet_max/(alf_max*hx + bet_max)
154     gamXmax[j] = gam_max/(alf_max + bet_max/hx)
155
156 N = (Nx+1)*(Ny+1) # no of unknowns
157 A = np.zeros((N, N))
158 # b = np.zeros((N))
159 b = [0]*N
160 m = lambda i, j: j*(Nx+1) + i
161 # Equations corresponding to j=0, i=0,1,... (u known)
162 j = 0

```

```

162 j = 0
163 for i in Ix:
164     p = m(i,j); A[p, p] = 1
165
166 # Loop over all internal mesh points in y direction
167 # and all mesh points in x direction
168 for j in Iy[1:-1]:
169     i = 0; p = m(i,j); A[p, p] = 1 # Boundary
170     for i in Ix[1:-1]: # Interior points
171         p = m(i,j)
172         A[p, m(i,j-1)] = - theta*Fy
173         A[p, m(i-1,j)] = - theta*Fx
174         A[p, p] = 1 + 2*theta*(Fx+Fy)
175         A[p, m(i+1,j)] = - theta*Fx
176         A[p, m(i,j+1)] = - theta*Fy
177     i = Nx; p = m(i,j); A[p, p] = 1 # Boundary
178 # Equations corresponding to j=Ny, i=0,1,... (u known)
179 j = Ny
180 for i in Ix:
181     p = m(i,j); A[p, p] = 1
182
183 import scipy.linalg
184 for n in It[0:-1]:
185     # Compute b
186     j = 0
187     for i in Ix:
188         p = m(i,j); b[p] = betYmin[i]*u_n[i][j] + gamYmin[i] # Boundary
189     for j in Iy[1:-1]:
190         i = 0; p = m(i,j); b[p] = betXmin[j]*u_n[i][j] + gamXmin[j] # Boundary
191         for i in Ix[1:-1]: # Interior points
192             p = m(i,j)
193             b[p] = u_n[i,j] + \
194                 (1-theta)*(
195                 Fx*(u_n[i+1,j] - 2*u_n[i,j] + u_n[i-1,j]) + \
196                 Fy*(u_n[i,j+1] - 2*u_n[i,j] + u_n[i,j-1])) + theta*dt*f[i][j] + (1-t
197         i = Nx; p = m(i,j); b[p] = betXmax[j]*u_n[i][j] + gamXmax[j] # Boundary

```

```

198     j = Ny
199     for i in Ix:
200         p = m(i,j); b[p] = betYmax[i]*u_n[i][j] + gamYmax[i] # Boundary
201
202     # Solve matrix system A*c = b
203     for i in range(len(b)):
204         if isnan(b[i] or isinf(b[i])):
205             b[i] = 0
206     c = scipy.linalg.solve(A, b)
207     # Fill u with vector c
208     for i in Ix:
209         for j in Iy:
210             u[i,j] = c[m(i,j)]
211
212     # Update u_n before next step
213     u_n, u = u, u_n
214     return u
215
216 def DP2(u_0,y1_init,y2_init,fy = free_y,Nx1=Nx1,Nx2=Nx2,Nt=Nt,Lx1=Lx1,Lx2=Lx2,T=T,d
217         d2=d2,nul=nul,nu2=nu2,r1=r1,r2=r2, eps = eps):
218     k = 0
219     backward = 0
220     u = np.zeros(1)
221     u[0] = u_0(0,0,0)
222     u_new = 0
223     y = np.zeros((2,2,Nx1+1,Nx2+1))
224     x1 = np.zeros(Nx1+1)
225     x2 = np.zeros(Nx2+1)
226     t = np.linspace(0+backward*T, T-backward*T, Nt+1)
227     v = 0 #0 or 1 v = v(x,t)
228     s3 = np.zeros((Nx1+1,Nx2+1))
229     zero = np.zeros((Nx1+1,Nx2+1))
230     lamb = np.zeros(1)
231     a = 0
232
233     lambd_left = 0; lambd_right = 1
234     #initialization of functions
235     y[0][0] = [[y1_init(i,j) for j in range(Nx2+1)] for i in range(Nx1+1)]
236     y[1][0] = [[y2_init(i,j) for j in range(Nx2+1)] for i in range(Nx1+1)]
237

```

```

198     j = Ny
199     for i in Ix:
200         p = m(i,j); b[p] = betYmax[i]*u_n[i][j] + gamYmax[i] # Boundary
201
202     # Solve matrix system A*c = b
203     for i in range(len(b)):
204         if isnan(b[i] or isinf(b[i])):
205             b[i] = 0
206     c = scipy.linalg.solve(A, b)
207     # Fill u with vector c
208     for i in Ix:
209         for j in Iy:
210             u[i,j] = c[m(i,j)]
211
212     # Update u_n before next step
213     u_n, u = u, u_n
214     return u
215
216 def DP2(u_0,y1_init,y2_init,fy = free_y,Nx1=Nx1,Nx2=Nx2,Nt=Nt,Lx1=Lx1,Lx2=Lx2,T=T,d
217         d2=d2,nul1=nul1,nu2=nu2,r1=r1,r2=r2, eps = eps):
218     k = 0
219     backward = 0
220     u = np.zeros(1)
221     u[0] = u_0(0,0,0)
222     u_new = 0
223     y = np.zeros((2,2,Nx1+1,Nx2+1))
224     x1 = np.zeros(Nx1+1)
225     x2 = np.zeros(Nx2+1)
226     t = np.linspace(0+backward*T, T-backward*T, Nt+1)
227     v = 0 #0 or 1 v = v(x,t)
228     s3 = np.zeros((Nx1+1,Nx2+1))
229     zero = np.zeros((Nx1+1,Nx2+1))
230     lamb = np.zeros(1)
231     a = 0
232
233     lambd_left = 0; lambd_right = 1
234     #initialization of functions
235     y[0][0] = [[y1_init(i,j) for j in range(Nx2+1)] for i in range(Nx1+1)]
236     y[1][0] = [[y2_init(i,j) for j in range(Nx2+1)] for i in range(Nx1+1)]

```

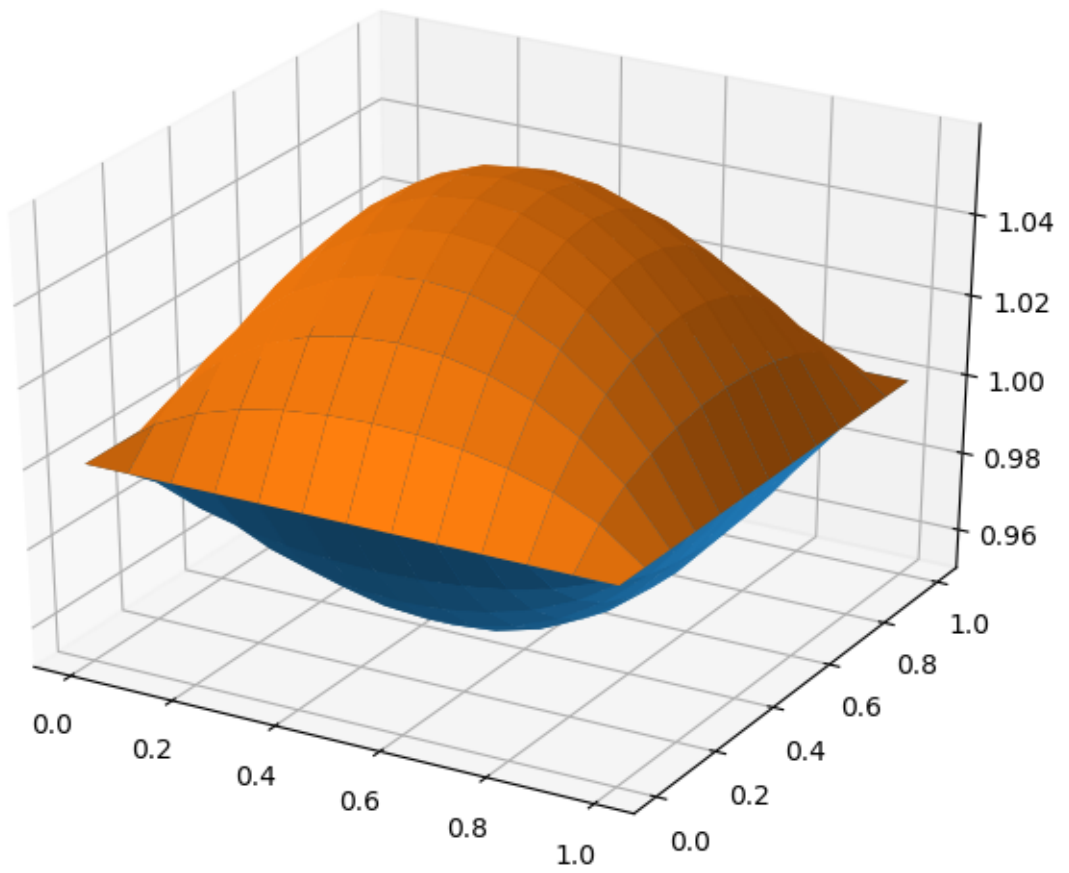
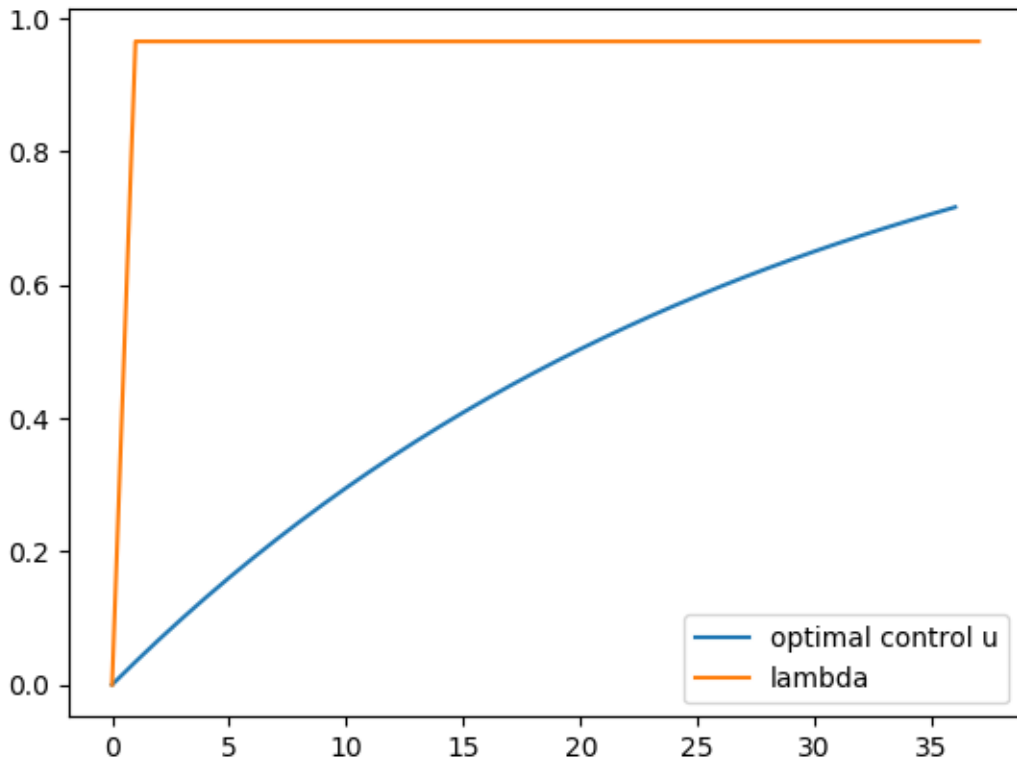
```

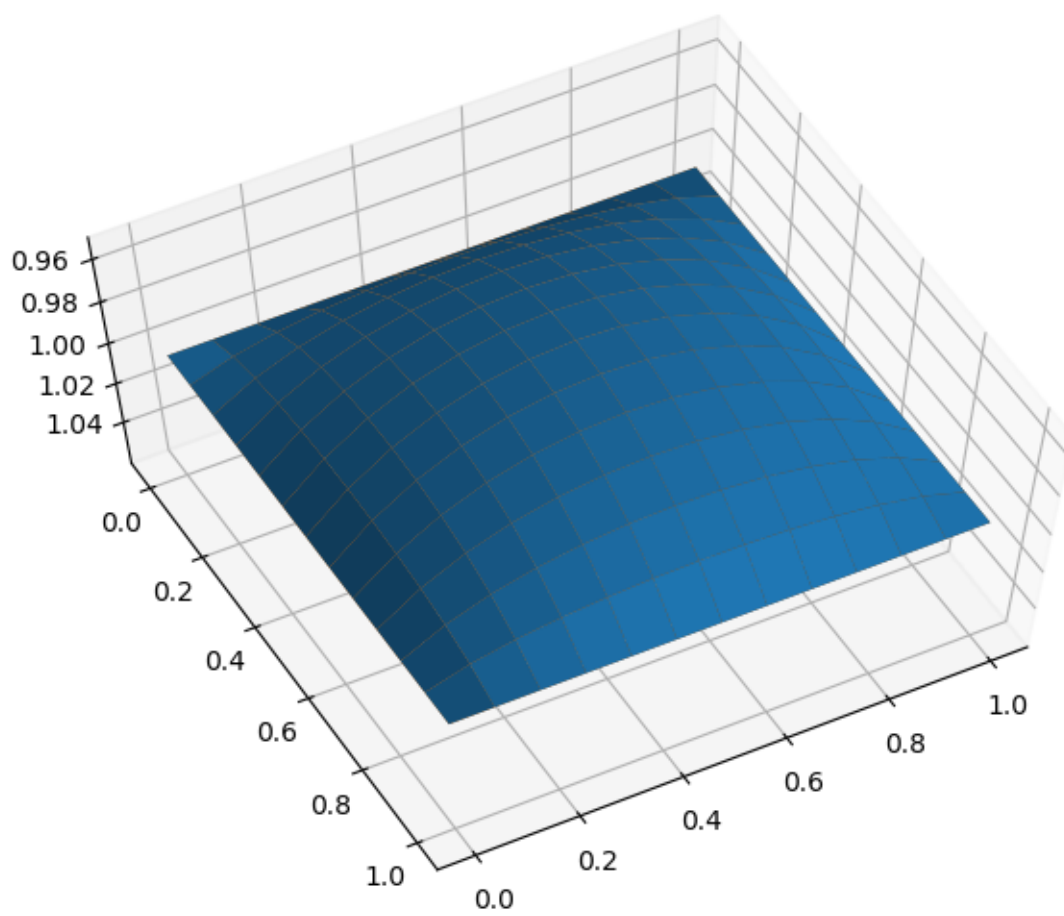
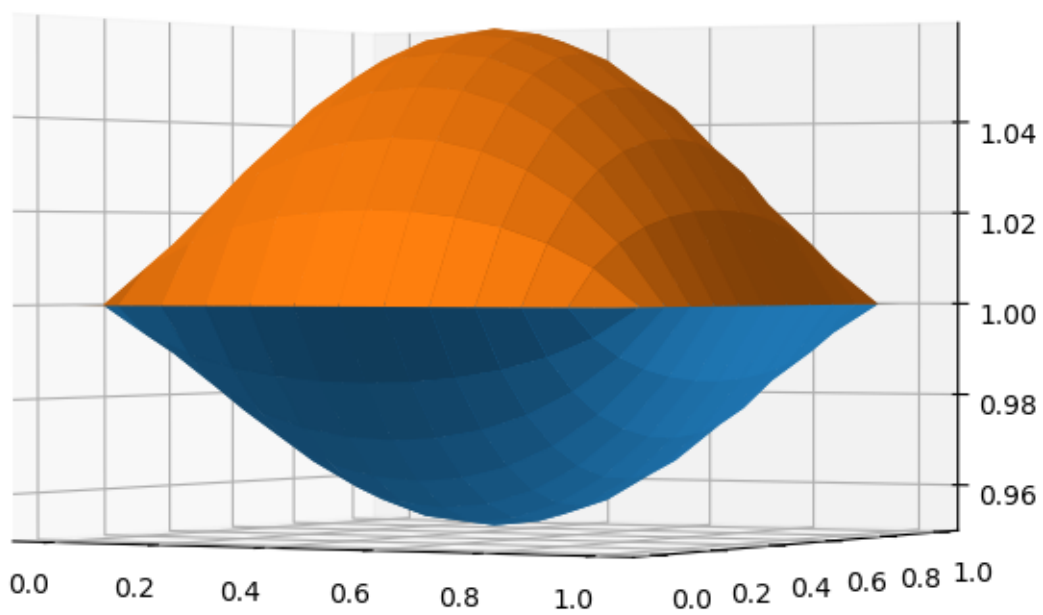
238 while True:
239     fy1 = fy(y[0][0],y[1][0],u[k],r2=0,nu2=0)
240     y[0][1] = solver_dense(y1_init,d1,fy1)
241
242     fy2 = fy(y[0][0],y[1][0],u[k],r1=0,nu1=0)
243     y[1][1] = solver_dense(y2_init,d2,fy2)
244
245     #S3
246     s3 = nu2*y[1][0] - nu1*y[0][0]
247     if np.all(s3 < zero) :
248         v = 0
249     else:
250         v = 1
251     #S4
252     s4_func = lambda a: a*u[k] + (1-a)*v
253     a = gold_bound(s4_func,lambd_left,lambd_right)
254     lamb = np.append(lamb,a)
255     #S5
256     u_new = a*u[k] + (1-a)*v
257     #S6
258     if (u_new - u[k]) < eps:
259         break;
260     else:
261         k += 1
262     u = np.append(u,u_new)
263     y[0][0],y[0][1] = y[0][1],y[0][0]
264     y[1][0],y[1][1] = y[1][1],y[1][0]
265
266     return (y[0][1],y[1][1],u,lamb)
267
268
269 x1,x2 = np.meshgrid(np.linspace(Lx1[0], Lx1[-1], Nx1+1), np.linspace(Lx2[0], Lx2[-1],
270 res = DP2(u,y1_init,y2_init)
271 plt.plot(res[2],label = 'optimal control u')
272 plt.plot(res[3], label = 'lambda')
273 plt.legend()
274 fig = plt.figure()
275 zyz = Axes3D(fig)
276 zyz.plot_surface(x1,x2,res[0],label = 'y1')
277 zyz.plot_surface(x1,x2,res[1],label = 'y2')
278 plt.show()

```

9 Отримані результати

Графік оптимального керування на певному проміжку та для певних початкових умов зазначених вище в програмі, побудованій в пакеті Python.





10 Висновки

Ця робота присвячена вивченню задачі оптимального управління динамічною моделлю популяції, пов'язаної з дифузійними моделями, тобто задача, яка регулюється диференціальними рівняннями в частинних похідних типу реакція-дифузія. Виведено умови існування оптимального керування та принцип максимуму, вказано числові алгоритми для наближення оптимальних значень функціоналів витрат.

Моє завдання полягало у створенні програми обчислення оптимального керування на одній із мов програмування – а саме Python. Представлена комп'ютерна програма, заснована на математичних результатах та побудована завдяки числовому наближенню методом градієнта.

11 Використана література

[1] Sebastian Anita, Viorel Arnautu, Vincenzo Capasso. An Introduction to Optimal Control Problems in Life Sciences and Economics. From Mathematical Models to Numerical Simulation with MATLAB. Springer – New York, Dordrecht, Heidelberg, London. – 2010. – P.247.

[2] Viorel Arnautu, Pekka Neittaanmäki. Optimal Control from Theory to Computer Programs. Springer Science+Business Media Dordrecht, Kluwer Academic Publishers in 2003. – 2003. – P.336.

[3] Джон Г. Мэтьюз, Куртис Д. Финк. Численные методы. Использование MATLAB - Numerical Methods: Using MATLAB. — 3-е изд. — М.: «Вильямс», 2001. — 720 с.