**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

Факультет Прикладної Математики та Інформатики

Кафедра Прикладної Математики та Інформатики

Магістерська робота

# Математичне моделювання поширення фейкових новин та пошук оптимальних параметрів

Виконав:
студент групи ПМПМ-22
спеціальності 113 - "Прикладна математика"
Школик Б.М.

Керівник:
__________ доц. Ящук Ю.О.
Рецензент:
__________ ____________________

Львів 2023

# Зміст

# Вступ

Кожен з нас використовує Інтернет як джерело новин або важливої для нас інформації, однак важко бути впевненим у правдивості цієї інформації. Як зазначено в [5], фейкові новини створюються шляхом повного ігнорування правил і процесів, які використовуються для забезпечення достовірності інформації. Вони можуть бути створені з різними цілями, але найпоширенішими є, звичайно, використання під час інформаційної війни з метою підриву якості інформації супротивника та виборчого процесу з метою дискредитації політичного опонента шляхом формування громадської думки. Інша мета - прибуток, який отримується в Інтернеті пропорційно кількості відвідувачів статті. Починаючи з цього моменту, стає важливим, наскільки швидко буде поширюватися неправдива інформація. Для цього можна використати добре досліджені в літературі моделі поширення епідемії.

Епідеміологічні моделі зазвичай представлені системою звичайних диференціальних рівнянь або рівнянь у частинних похідних. SIR модель є математичною моделлю, яка використовується для моделювання поширення інфекційних захворювань у населенні, але її також можна використовувати для моделювання поширення фейкових новин. Модель складається з трьох категорій: Susceptible (схильні до інфекції), Infected (інфіковані) та Recovered (відновлені або вилікувані).

У цій роботі ми зосередимося на переформулюванні моделі SIR для отримання моделі поширення фейкових новин. Також ми розглянемо задачу оптимального керування та пошук опимальних параметрів для неї.

Для моделювання поширення фейкових новин, "схильні до інфекції" - люди, які є потенційними авторами фейкових новин, "інфіковані" - активні автори, які розміщують фейкову інформацію, а "відновлені" - автори, які неактивні до поширення фейкових новин [7].

# 1 Математичне моделювання фейкової інформації

## 1.1 Визначення неправдивої інформації

В сучасному світі Інтернет відіграє важливу роль. Він надає широкий доступ до інформації на різні теми, а також дає можливість комунікувати з іншими людьми з усього світу. Однак, разом з цим, з'явилася також проблема неправдивої інформації, яка може негативно впливати на думку та рішення людей. Неправдива інформація (також відома як "фейкові новини") - це інформація, яка містить неправдиві або маніпулятивні твердження та призначена для введення людей в оману. Це може бути різного типу інформація, наприклад, викривлені факти, чутки, пропаганда або навіть повністю вигадані історії.

Згідно з [11], неправдиву інформацію поділяють на:

- **дезінформація** - інформація, яка є неправдивою та навмисно створена, щоб завдати шкоди людині, соціальній групі, організації чи країні
- **мізінформація** - інформація, яка є неправдивою, але не створена з наміром заподіяти шкоду
- **малінформація** - інформація, яка базується на реальності, використовується для заподіяння шкоди особі, соціальній групі, організації чи країні.

Така інформація може мати серйозні наслідки для суспільства, такі як вплив на вибори, настрої та переконання громадян, а також може негативно впливати на економіку та міжнародні стосунки. Часто неправдива інформація поширюється швидко через соціальні мережі та інші онлайн-канали, тому поширення важко контролювати.

Тому, моделювання поширення фейкових новин має значення з точки зору прогнозування, як швидко та в якому обсязі фейкові новини можуть поширюватися в мережі, що допомагає виявляти й запобігати небажаним наслідкам цього поширення. Також пошук оптимальних значень моделі дає змогу зрозуміти як розробляти ефективні антифейкові стратегії, наприклад, шляхом зменшення кількості людей, які бачать фейкові новини або підвищення рівня освіти та критичного мислення.

## 1.2 Поширення фейкових новин

Поширення фейкових новин є серйозною проблемою в сучасному світі. Оскільки фейки можуть швидко поширюватися в Інтернеті та соціальних медіа, виявлення фейкових новин є нелегким завданням. Один зі способів виявлення фейкових новин - це використання інформаційних технологій та аналізу даних. Однак, наявність багатьох типів фейкових новин важко узагальнити під одну методику.

Кожен тип фейку може потребувати окремого підходу для їх виявлення. Деякі типи фейкових новин можуть бути виявлені шляхом перевірки фактів та дослідження джерел, що їх публікують. Інші можуть бути виявлені шляхом аналізу мовлення та інших характеристик контенту. Ще інші можуть вимагати використання більш складних технологій, таких як аналіз зображень та відео.

Виявлення фейкових новин може залежати від контексту та культурного середовища, в якому вони з'являються. Наприклад, фейки про політичних лідерів можуть бути більш поширеними під час виборчих кампаній, а фейки про наукові дослідження можуть бути поширеними під час важливих міжнародних конференцій. Усі ці фактори створюють потребу в постійному дослідженні та розробці нових методів виявлення фейкових новин. Такі методи можуть включати як традиційні, так і нові підходи, такі як використання штучного інтелекту та машинного навчання.

В існуючій літературі багато авторів намагалися надати математичні моделі для розповсюдження фейкових новин, і поки що математичні моделі в епідеміології є найкращими для вивчення поширення фейкових новин, оскільки вони здатні описати реалістичну взаємодію між фейковою інформацією та людьми, які її отримують [6], [8],[9] .

Класичні математичні моделі для епідеміології досліджуються з метою створення більш реалістичних інструментів, здатних прогнозувати поширення захворювань. Наприклад, модель SIR для опису фейкової інформації наведена в [7], де автори надають аналіз жорсткості для поширення фейкових новин. Зокрема, цей коефіцієнт використовується для розуміння того, наскільки швидко відбувається передача фейкових новин у певній популяції на основі реальних даних.

### 1.3 SIR-модель для поширення фейкових новин

Як зазначалося вище, у [7] автори переробили модель SIR адаптуючи її для фейкових новин:

- S(t): потенційні автори фейкових новин;
- I(t): активні автори, які розміщують фейкову інформацію;
- R(t): неактивні автори фейкових новин.

Далі вони подали її у вигляді системи звичайних диференціальних рівнянь:

$$\begin{cases} \dfrac{dS}{dt} = -\beta S(t)I(t) \\ \dfrac{dI}{dt} = \beta S(t)I(t) - \alpha I(t) \\ \dfrac{dR}{dt} = \alpha I(t) \end{cases} \tag{1}$$

де $\alpha$ – швидкість відновлення, а $\beta$ – швидкість контакту. Зокрема, ці параметри пов'язані з двома важливими індексами, які зазвичай використовуються для опису соціальних, економічних і культурних результатів нашого суспільства та можуть бути різними для кожної країни, що дає змогу порівняти вплив фейкових новин. Індекс проникнення Інтернету $i$ та індекс людського розвитку $h$ використовуються для подання параметрів моделі, а саме:

$$\beta = \frac{i}{10}, \quad \alpha \quad = \frac{h}{100}.$$

Значення цих параметрів для вибраних країн наведені у звіті "Програми розвитку ООН" на 2022 рік наведено в Табл. 1.

Табл. 1: Значення констант $\alpha$, $\beta$ в (1) для Франції, Індії, Італії, Мексика та Сполучених Штатів, посилаючись на 2022 рік.

| Країна | $\alpha$ | $\beta$ |
|---|---|---|
| Франція | 0.009 | 0.092 |
| Індія | 0.006 | 0.060 |
| Італія | 0.009 | 0.085 |
| Мексика | 0.008 | 0.071 |
| США | 0.009 | 0.091 |

Для програмної реалізації було використано функцію ode15s [1] на платформі Matlab.

### 1.4 Числові результати

Табл. 2: Кількість одиниць часу, необхідних для досягнення максимальної кількості заражених у Франції, Індії, Італії, Мексиці та США, посилаючись на 2022 рік.

| Країна | Кількість одиниць часу |
|---|---|
| Франція | 61.8074 |
| Індія | 94.5629 |
| Італія | 66.4362 |
| Мексика | 79.2521 |
| США | 62.4165 |

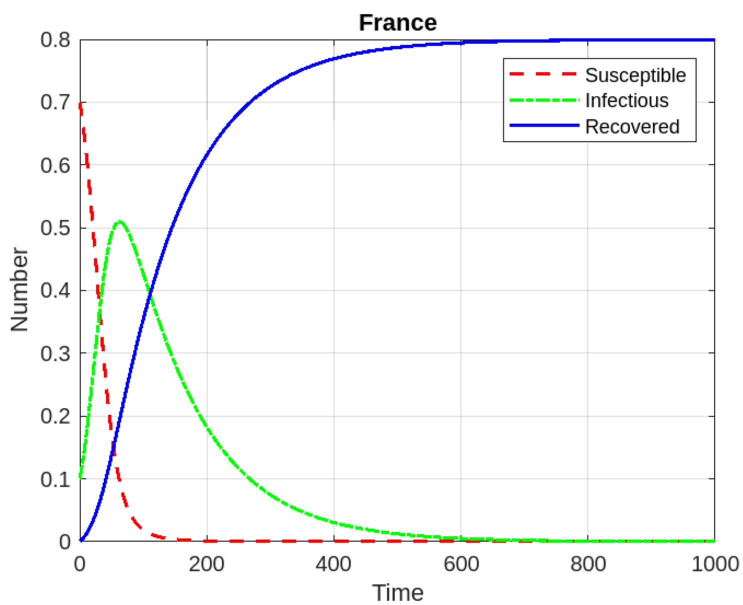


Рис. 1: Розв'язок моделі SIR (1) для Франції.

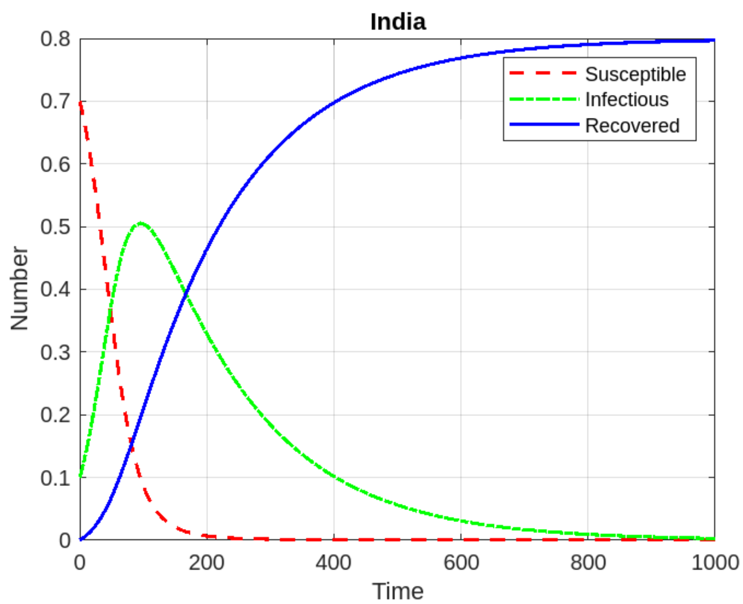


Рис. 2: Розв'язок моделі SIR (1) для Індії.

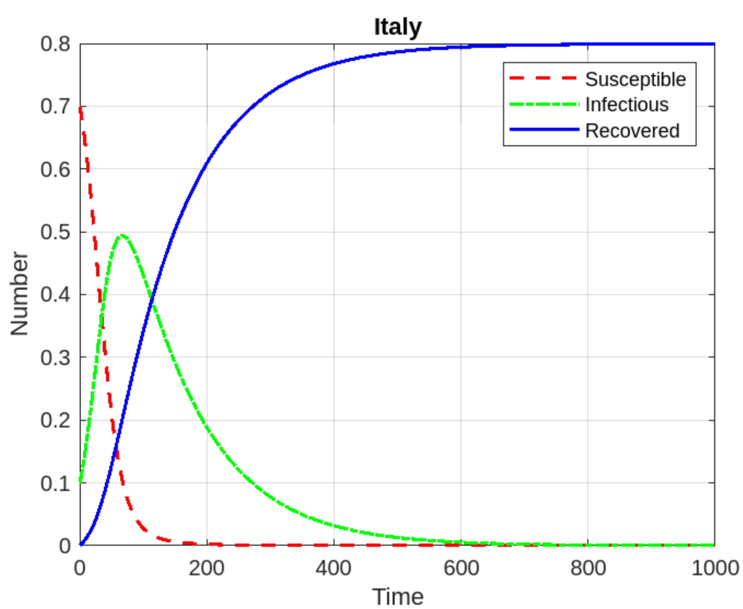


Рис. 3: Розв'язок моделі SIR (1) для Італії.

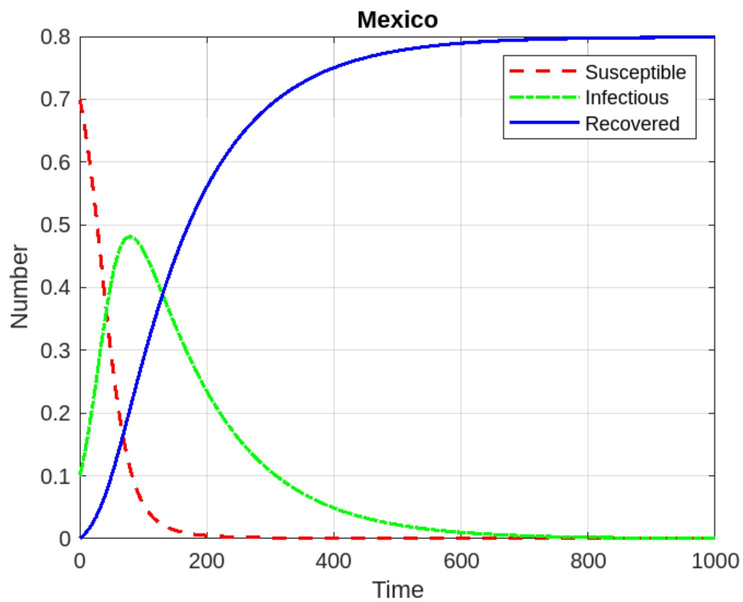


Рис. 4: Розв'язок моделі SIR (1) для Мексики.

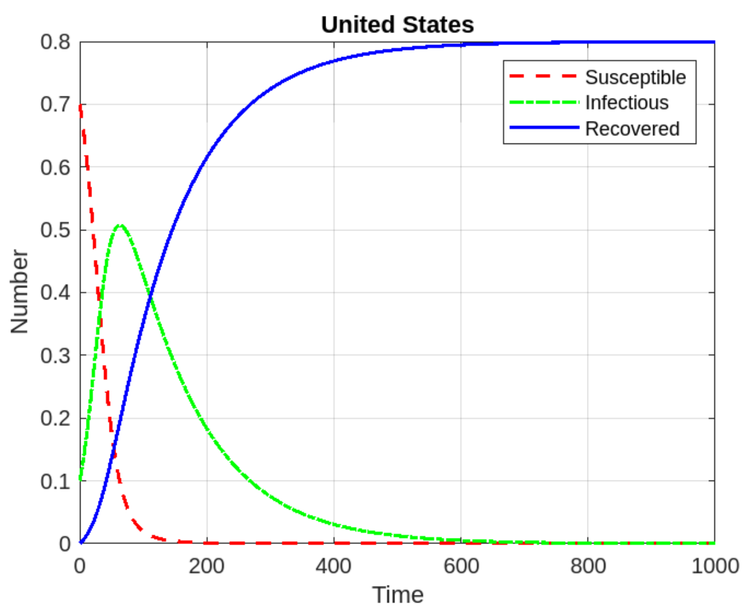


Рис. 5: Розв'язок моделі SIR (1) для США.

## 2 Задача оптимального керування

[10], [4] Оптимізація є важливим інструментом у науці прийняття рішень та в аналізі фізичних систем. Щоб скористатися цим інструментом, ми повинні спочатку визначити певну цільову функцію, яку потрібео оптимізувати. Це може бути прибуток, час, потенціальна енергія або комбінація величин, які можуть бути представлені одним числом. Цільова функція залежить від певних характеристик системи, які називаються змінними або невідомими. Наша мета — знайти значення змінних, які оптимізують обрану функцію. Часто змінні можуть бути певним чином обмежені. Наприклад, такі величини, як електронна густина в молекулі та відсоткова ставка за кредитом, не можуть бути від'ємними.

Існує дуже багато застосувань, які можна сформулювати як проблеми безперервної оптимізації; наприклад:

- знаходження оптимальної траєкторії руху літака або руки робота;
- проектування портфеля інвестицій для максимізації очікуваного прибутку при збереженні прийнятного рівня ризику;
- керування хімічним процесом або механічним пристроєм для оптимізації продуктивності або відповідності стандартам міцності;
- обчислення оптимальної форми компонента автомобіля або літака.

Процес визначення цільової функції, змінних і обмежень для даної проблеми відомий як моделювання. Побудова відповідної моделі є першим, іноді найважливішим кроком у процесі оптимізації. Якщо модель надто спрощена, вона не дасть корисного розуміння практичної проблеми. Якщо вона надто складна, її може бути надто важко вирішити.

Після того, як модель сформульована, алгоритм оптимізації може бути використаний для пошуку її рішення, зазвичай за допомогою комп'ютера. Немає універсального алгоритму оптимізації. Для пошуку використовують набір алгоритмів, кожен з яких адаптований до певного типу проблеми оптимізації. Вибір алгоритму, який підходить для конкретної програми, здійснюється людиною.

Після того, як алгоритм оптимізації було застосовано до моделі, ми повинні бути в змозі розпізнати, чи вдалося йому виконати поставлене завдання. У багатьох випадках існують елегантні математичні вирази, відомі як умови оптимальності для перевірки того, що поточний набір змінних справді є рішенням задачі. Якщо умови оптимальності не задовольняються, вони можуть дати корисну інформацію про те, як можна покращити рішення. Модель можна покращити шляхом застосування таких методів, як аналіз чутливості, який виявляє чутливість рішення до змін у моделі. Якщо в модель вносяться будь-які зміни, задача оптимізації вирішується заново, і процес повторюється.

Математичною моделлю системи є оператор $c$, який дозволяє по множині заданих вхідних параметрів $P$ знаходити значення вихідних параметрів $y$, які цікавлять дослідника.

Набір вхідних параметрів $P = (p_v, p_s, p_0)^T$:

- $p_v$ - внутрішні параметри системи;
- $p_s$ - параметри впливу зовнішнього середовища (граничні умови);
- $p_0$ - початкові умови.

Набір параметрів $P$ поділяють на дві групи: $P = [p, u]^T$

- $p$ - фіксовані (постійні) параметри;
- $u$ - параметри оптимізації.

Оскільки ми розглядали задачу оптимізації (наше завдання полягає в знаходженні значень оптимізаційних параметрів), модель системи можна подати у наступному вигляді:

$$c(y, u) = 0 \tag{2}$$

Модель (2) може бути представлена у вигляді системи алгебраїчних рівнянь, звичайних диференціальних рівнянь, рівнянь у часткових похідних еліптичного, параболічного або гіперболічного типів тощо. Рівняння (2), як правило, розв'язується відносно $y$ і не може бути розв'язане відносно $u$. Підставивши значення $u$ і отримавши з рівняння (2) значення $y$, можна обчислити значення функціоналів $\psi_j, j = \overline{0, m}$.

Будь-яка задача оптимізації потребує наступних компонентів:

- Змінні оптимізації (також називають змінними дизайну, змінними керування, функцією керування, керуванням) позначаються як вектор $u, u \in U, U$ - множина керувань;
- Функція стану (стан, функція поведінки) $y, y \in Y, Y$ - множина станів;
- Модель системи. Для заданого керування $u$, стан системи $y(u)$ отримується як результат розв'язання рівняння (2) або (3).

$$c(y, u, p) = 0 \tag{3}$$

  Також (2), (3) називають прямою задачею (задачею аналізу).

- Функція витрат (її також називають цільовою функцією/функціоналом, функціоналом витрат, критерієм оптимізації, критерієм витрат), позначають як $\psi_0(u) = \tilde{\psi}_0(u, y)$.

- Обмеження (функціонали обмеженнь), виражені як рівності або нерівності, позначаються як

$$\psi_j(u) = \tilde{\psi}_j(u, y) = 0, \quad j = 1, ..., m_1,$$
$$\psi_j(u) = \tilde{\psi}_j(u, y) \leq 0, \quad j = 1, ..., m_2,$$
$$m_1 + m_2 = m.$$

Для компактності подання формул можна записати всі обмеження як:

$$\tilde{\psi}_j(u, y) \leq 0, \quad j = 1, ..., m,$$

Слід зауважити, що кожне обмеження рівності можна представити у вигляді двох обмежень нерівності. Крім того, обмеження нерівності можна перетворити на обмеження рівності.

## 2.1 Постановка задачі оптимального керування

Нехай $\mathbf{p} = [\alpha, \beta]^T$ та $iu$ - індекс параметру цього вектору для наших рівнянь. Параметр моделі $iu = 2$ обрано як функцію керування:

$$u(t) = p_1 = p(1),$$
$$u^- \leq u(t) \leq u^+.$$

Функція керування апроксимується кусково-сталою функцією:

$$u(t) = u(t, b), \quad b = [b_1, ..., b_n]^T.$$

Давайте перепишемо нашу модель наступним чином:

$$y = [S, I, R]^T = [y_1, y_2, y_3]^T \in Y$$

тоді

$$c(y, u) = \begin{pmatrix} \begin{cases} y_1'(t) + \beta y_1(t) y_2(t) \\ y_2'(t) - \beta y_1(t) y_2(t) + \alpha y_2(t) \\ y_3'(t) - \alpha y_2(t), \\ t \in (t_0, T] \end{cases} \\ y_1(t_0) - y_{10}, \\ y_2(t_0) - y_{20}, \\ y_3(t_0) - y_{30}. \end{pmatrix} = 0, \tag{4}$$

де $y_{10}, y_{20}$ та $y_{30}$ - початкові значення. Інтервал $[t_0, T]$ ділиться на $n$ рівних підінтервалів.

В даній задачі будемо розглядати критерій оптимізації як: знайти таку керуючу функцію $u(t) = p(1)$, щоб максимальне значення інфікованих (див. [I(t) з моделі]), що розглядається на проміжку $(t_0, T]$ було мінімальним. Тоді наш критерій оптимізації можемо записати як:

$$\psi_0(u) = \tilde{\psi}_0(u, y) = \max y_2(t)$$

Також запишемо обмеження для інфікованих (див. [I(t) з моделі]) як:

$$y_2(t) \leq y_{2u}(t),$$

де $y_{2u}(t)$ - верхня межа для інфікованих.

Дану нерівність можемо перетворити на інтегральне обмеження рівності:

$$\psi_1(u) = \tilde{\psi_1}(u, y) = \int_{t_0}^{T} \left( (y_2(t) - y_{2u}(t)) + |y_2(t) - y_{2u}(t)| \right)^2 dt$$

Задачі оптимального керування за своєю суттю є нескінченним задачами, оскільки ми шукаємо рішення у функціональних просторах. Ми можемо поділити числові методи для цих задач на два основні класи: прямі (дискретизувати, а потім оптимізувати) і непрямі (оптимізувати, а потім дискретизувати) методи.

У непрямих методах ми спочатку виводимо умови оптимальності у функціональному просторі, які потім дискретизуємо. У прямих методах ми спочатку дискретизуємо задачу, а потім знаходимо оптимізатор отриманої задачі нелінійного програмування.

Однією з переваг прямих методів є те, що умови оптимальності задач нелінійного програмування є загальними, тоді як умови оптимальності недискретизованих задач оптимального керування потрібно перевизначати для кожної нової задачі та часто вимагають часткового апріорного знання математичної структури рішення, яке загалом не є доступний для багатьох прикладних проблем.

Неперервне формулювання для нашої задачі буде мати вигляд:

$$\begin{aligned}
U &= \left\{ u : u^- \leq u(t) \leq u^+ \right\} \\
\tilde{U_d} &= \left\{ (u, y) : u \in U, \quad y \in Y, \quad c(y, u) = 0, \quad \tilde{\psi_1}(u, y) = 0 \right\} \\
U_d &= \left\{ u : u \in U, \quad \tilde{\psi_1}(u) = 0 \right\}
\end{aligned}$$

Задача оптимізації (розширена форма):
знайти такі функції $u_*$, $y_*$, що

$$(u_*, y_*) \in \tilde{U_d}, \quad \tilde{\psi_0}(u_*, y_*) = \min_{(u,y) \in \tilde{U_d}} \tilde{\psi_0}(u, y)$$

Задача оптимізації (скорочена форма):
знайти таку функцію $u_*$, що

$$u_* \in U_d, \quad \psi_0(u_*) = \min_{u \in U_d} \psi_0(u), \quad c(y, u) = 0$$

Розглянемо прямий метод вирішення задачі оптимізації – спочатку дискретизувати, потім оптимізувати. Тоід задачу оптимізації можна записати як задачу нелінійного програмування:

$$U_b = \Big\{u : b \in R^n, \quad b^- \le b \le b^+, \quad b^- \in R^n, \quad b^+ \in R^n\Big\}$$
$$\tilde{U_d} = \Big\{b : b \in U_b, \quad y \in Y, \quad \tilde{\psi_1}(b) \equiv \tilde{\psi_1}(y, b) = 0, \quad c(y, b) = 0\Big\}$$
$$U_d = \Big\{b : u \in U_b, \quad \tilde{\psi_1}(b) \equiv \tilde{\psi_1}(y(b), b) = 0\Big\}$$

Задача оптимізації (розширена форма):
знайти такі функції $b_*$, $y_*$, що

$$(b_*, y_*) \in \tilde{U_d}, \quad \tilde{\psi_0}(b_*, y_*) = \min_{(b,y)\in\tilde{U_d}} \tilde{\psi_0}(b, y)$$

Задача оптимізації (скорочена форма):
знайти такий вектор $b_*$, що

$$b_* \in U_d, \quad \psi_0(b_*) = \min_{b\in U_d} \psi_0(b), \quad c(y, b) = 0$$

## 2.2 Програмна реалізація

Програмна реалізація для даної задачі була виконана на платформі Matlab. Програма складається з модулів:

- `app1.mlapp`;
- `fake_news_constr.m`;
- `fake_news_optcr.m`;
- `fake_news_scr_optim.m`;
- `fake_news.m`.

Давайте розглянемо, що робить кожен модуль:

Модуль `app1.mlapp` виконує відображення інтерфейсу користувача для зручності використання програми. Реалізацію здійснено за допомогою Matlab App Designer [2].

Модуль `fake_news_constr.m` обраховує значення обмеження.

Модуль `fake_news_optcr.m` обраховує значення критерію оптимізації.

Модуль `fake_news_scr_optim.m` - це головна точка входу програми. В даному модулі для пошуку оптимальних параметрів було використано вбудовану функцію fmincon [3].

Модуль `fake_news.m` описує нашу модель (1).

## 2.3 Числові результати

Числові значення вхідних даних:

$[t_0, T] = [0, 1000]\,;$
$[y_{10}, y_{20}, y_{30}] = [0.7, 0.1, 0]\,;$
$p = [0.009, 0.090]\,;$

Нижня та верхня межі для параметрів моделі:

$p_l = [0.002, 0.050]\,;$
$p_u = [0.009, 0.091]\,;$

Табл. 3: Значення критерію оптимізації та обмеження для початкового значення параметрів оптимізації $b_0$.

| $\psi_0(b_0)$ | $\psi_1(b_0)$ |
|---|---|
| 0.5050 | 26.0789 |

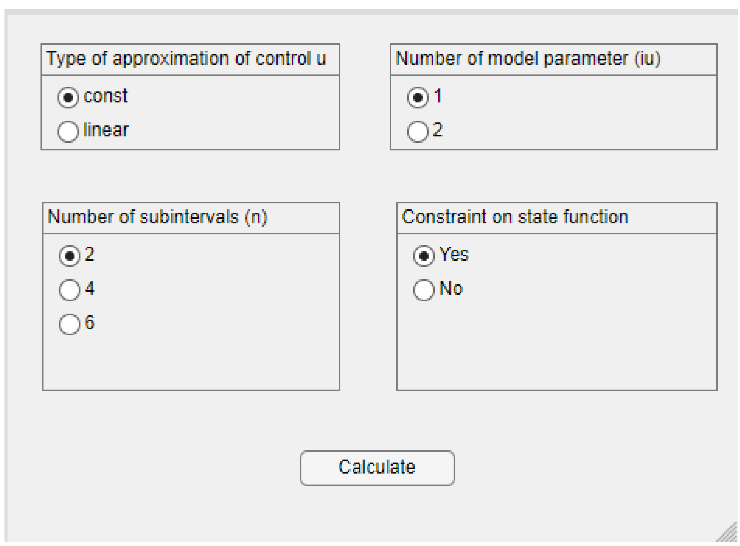


Рис. 6: Інтерфейс користувача.

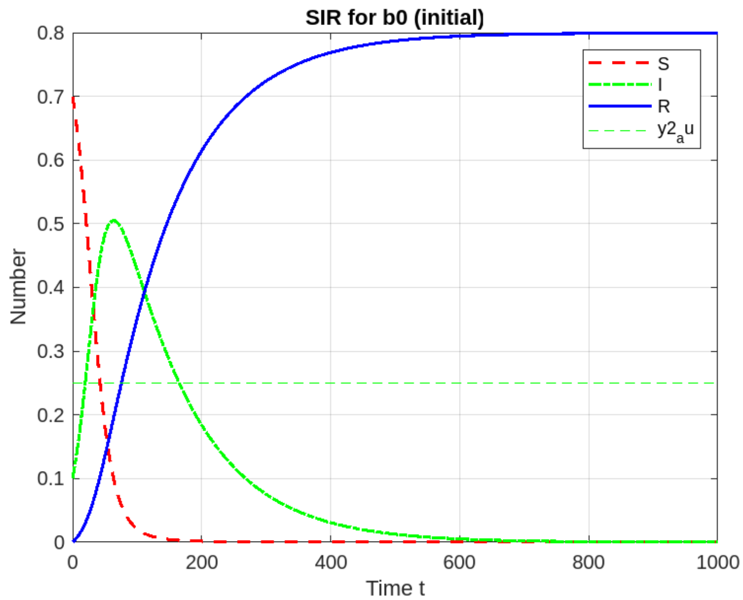


Рис. 7: Графіки моделі SIR для початкового значення параметрів оптимізації $b_0$.

Табл. 4: Оптимальні значення параметрів оптимізації, критеріїв оптимізації та обмеження $\psi_1$ для різної кількості підінтервалів $n = 2, 4, 6$ (без обмеження $\psi_1(u)$ ).

| $n$ | $b_*$ | $\psi_0(b_*)$ | $\psi_1(b_*)$ |
|---|---|---|---|
| 2 | 0.0500<br>0.0900 | 0.3754 | 4.5415 |
| 4 | 0.0500<br>0.0900<br>0.0900<br>0.0900 | 0.3754 | 4.5380 |
| 6 | 0.0500<br>0.0900<br>0.0900<br>0.0900<br>0.0900<br>0.0900 | 0.3754 | 4.5658 |

Табл. 5: Оптимальні значення параметрів оптимізації, критеріїв оптимізації та обмеження $\psi_1$ для різної кількості підінтервалів $n = 2, 4, 6$ (з обмеження $\psi_1(u)$ ).

| $n$ | $b_*$ | $\psi_0(b_*)$ | $\psi_1(b_*)$ |
|---|---|---|---|
| 2 | 0.0500<br>0.0900 | 0.3754 | 4.5415 |
| 4 | 0.0500<br>0.0900<br>0.0900<br>0.0900 | 0.3754 | 4.5380 |
| 6 | 0.0500<br>0.0500<br>0.0900<br>0.0900<br>0.0900<br>0.0900 | 0.3754 | 4.5380 |

Наведемо графіки для $n = 6$.

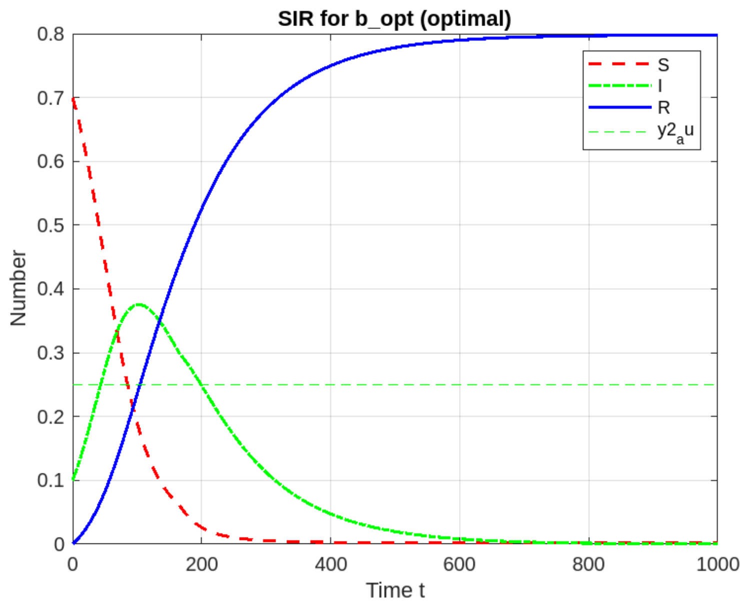


Рис. 8: Графіки моделі SIR для оптимального значення параметрів оптимізації $b_*$ (без обмеження $\psi_1$).

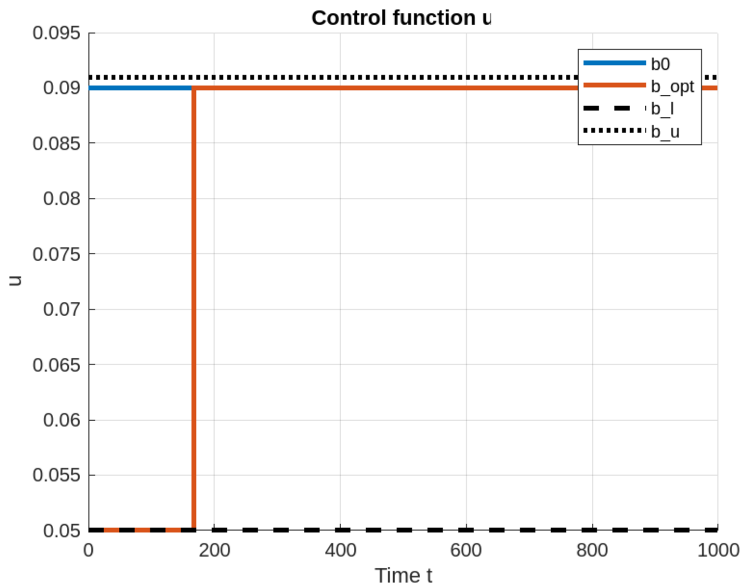


Рис. 9: Графіки функції керування $u(t)$ (початковий, оптимальний, нижня та верхня межі) (без обмеження $\psi_1$).

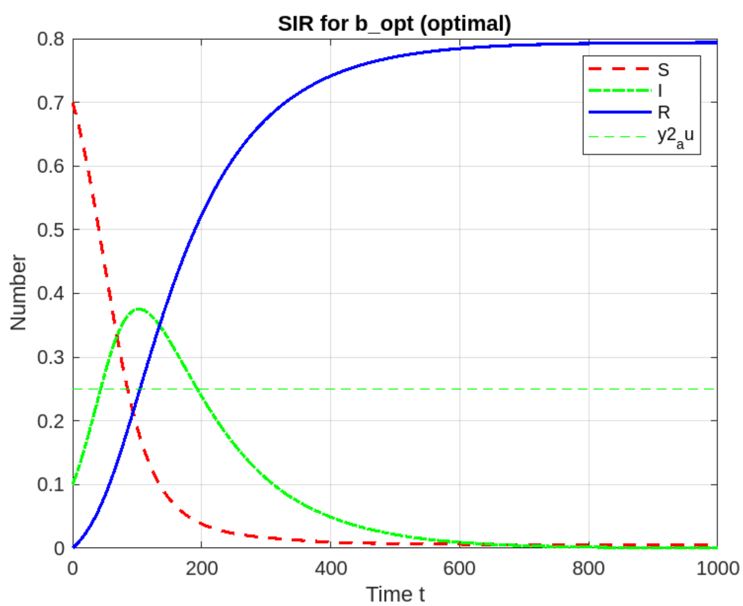


Рис. 10: Графіки моделі SIR для оптимального значення параметрів оптимізації $b_*$ (з обмеження $\psi_1$).

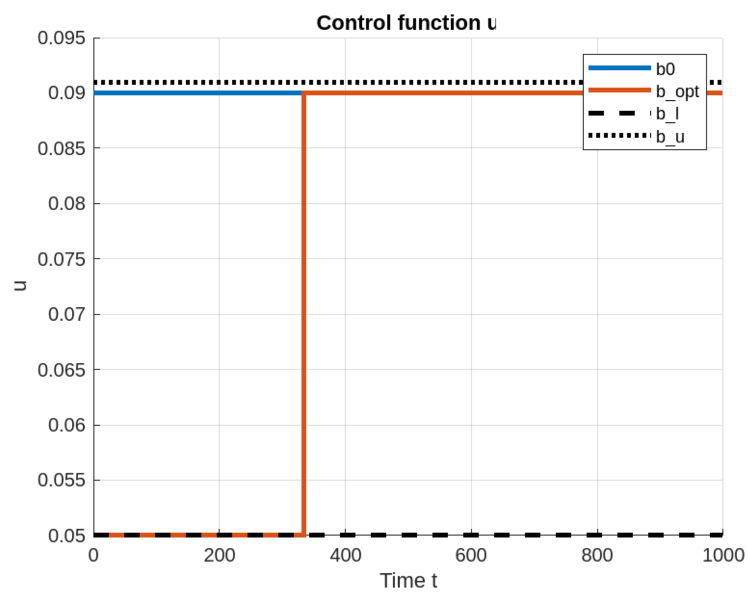


Рис. 11: Графіки функції керування $u(t)$ (початковий, оптимальний, нижня та верхня межі) (з обмеження $\psi_1$).

# Висновки

У цій роботі розглядалася модель SIR для моделювання поширення фейкової інформації. Також сформульовано задачу оптимального керування для нашої моделі. Для розв'язування використовувався прямий метод. Реалізація програмного забезпечення здійснюється за допомогою вбудованих функцій на платформі Matlab (безкоштовна пробна версія). Для створення інтерфейсу користувача використовувався Matlab App Designer. Щоб вдосконалити модель можна ввести нові параметри, які безпосередньо впливають на поширення фейкових новин, що дозволить отримувати більш реалістичні дані.

## Література


[1] https://www.mathworks.com/help/matlab/ref/ode15s.html.

[2] https://www.mathworks.com/products/matlab/app-designer.html.

[3] https://www.mathworks.com/help/optim/ug/fmincon.html.

[4] Jasbir Arora. *Introduction to Optimum Design.* Academic Press, 2016.

[5] Dorje C. Brody and David M. Meier. Mathematical models for fake news. *arXiv e-prints*, page arXiv:1809.00964, September 2018.

[6] Suyalatu Dong, Yan-Bin Deng, and Yong-Chang Huang. Seir model of rumor spreading in online social network with varying total population size. *Communications in Theoretical Physics*, 68:545, 10 2017.

[7] Raffaele D'Ambrosio, Giuseppe Giordano, Serena Mottola, and Beatrice Paternoster. Stiffness analysis to predict the spread out of fake information. *Future Internet*, 13(9), 2021.

[8] Hosam Mahmoud. A model for the spreading of fake news. *Journal of Applied Probability*, 57(1):332–342, 2020.

[9] Taichi Murayama, Shoko Wakamiya, Eiji Aramaki, and Ryota Kobayashi. Modeling the spread of fake news on twitter. *PloS one*, 16:e0250419, 04 2021.

[10] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, New York, NY, USA, 2e edition, 2006.

[11] United Nations Educational, Scientific and Cultural Organization (UNESCO). Fight fake news. https://en.unesco.org/fightfakenews.

# Додаток

## Програма 1

```
function res = country_sir_constants(countryName)
    switch countryName
        case 'France'
            res = [0.009 0.089];
        case 'India'
            res = [0.006 0.060];
        case 'Italy'
            res = [0.009 0.085];
        case 'Mexico'
            res = [0.008 0.071];
       case 'United States'
            res = [0.009 0.091];
    end
end

function res = sir_model(t, y)
    % y = [S I R]
    global alpha beta;

    res = [
        -beta*y(1)*y(2)
        beta*y(1)*y(2) - alpha*y(2)
        alpha*y(2)
        ];
end

function src_sir_model(countryName)

global alpha beta time;
global S0 I0 R0;

disp('--------------------------------');
disp('SIR model')
disp('--------------------------------');

disp('--------------------------------');
disp('Alpha, Beta')
disp('--------------------------------');
res = country_sir_constants(countryName);
alpha = res(1)
```

```
beta = res(2)

disp('----------------------------------------');
disp('Time')
disp('----------------------------------------');
time = 1000

disp('----------------------------------------');
disp('Initial values S(0), I(0) and R(0)')
disp('----------------------------------------');
S0=0.7
I0=0.1
R0=0

% res = [S I R]
[t, res]=ode15s(@sir_model,[0 time],[S0 I0 R0]);

disp('----------------------------------------');
disp('Number of time units required to reach the maximum of infected')
disp('----------------------------------------');

[maxValue, index] = max(res(:,2));
timeUnitsNumber = t(index)

figure
p = plot(t,res(:, 1), ...
    t,res(:, 2), ...
    t,res(:, 3));

p(1).LineWidth = 1.5;
p(2).LineWidth = 1.5;
p(3).LineWidth = 1.5;

p(1).Color = "red";
p(2).Color = "green";
p(3).Color = "blue";

p(1).LineStyle = "--";
p(2).LineStyle = "-.";
p(3).LineStyle = "-";

title(countryName);
xlabel('Time');
ylabel('Number');
legend('Susceptible','Infectious','Recovered')
grid on
```

```
end
```

**Програма 2**

```
function [c, ceq] = fake_news_constr(b_current)
  %Calculation value of constraint

  global x y0
  global nsol_ode

  global y
  global b1 y2_au %constaints

  global options_ode

  c=[];

  if abs(dot(b_current - b1, b_current - b1)) > eps
    [x, y] = ode15s(@fake_news, x, y0, options_ode);
    nsol_ode = nsol_ode + 1;
  end

  ceq = trapz(x, ((y(:, 2) - y2_au) + abs((y(:, 2) - y2_au))).^2);
end

function [optcr_1] = fake_news_optcr(b_current)
    %Calculation value of optimization criterion

    global x y0
    global nsol_ode

    global y
    global b1 %constaints

    %------------------------------------------------------------
    global b
    global options_ode

    b=b_current;


    %solution ODEs
    [x,y] = ode15s(@fake_news,x,y0,options_ode);
```

```
    %calculation value of optimization criterion
    optcr_1 = max(y(:, 2));

    nsol_ode=nsol_ode+1;
    b1 = b_current;
end

function fake_news_scr_optim(typeUInput, ...
    constraintInput, ...
    iuInput, ...
    nInput ...
)
    global p
    global iu
    global t0_te x y0
    global nsol_ode

    % ---------------------------------
    global n lts b typeU
    global options_ode
    %---------------------------------

    %---------------------------------
    global y
    global y2_au %constaints
    %---------------------------------

    constraint = constraintInput;

    disp('Fake news SIR model')
    disp('------------------------------------');

    disp('Input parameters')
    disp('Time interval')
    t0_te=[0 1000]; %time interval [0 1000]
    disp('------------------------------------');
    disp('Initial values [S(0), I(0), R(0)]')
    disp('------------------------------------');
    y0 = [0.7, 0.1, 0]

    %Model parameters (initial values)
    disp('Model parameters (initial values)')
    p=[0.009 0.090];

    %Lower and upper bounds for model parameters
    disp('Lower and upper bounds for model parameters')
```

```
p_l=[0.002  0.050];
p_u=[0.009  0.091];

%----------------------------------------

%Index of optimization parameters u in the vector p
disp('Index iu of optimization parameter in the vector p')
iu=iuInput

disp('Number of subintervals n')
n=nInput

%Type of approximation of control u
disp('Type of approximation of control u')
typeU = typeUInput;

y2_au = 0.25;

disp('-----------------------------------------------------')
disp('')

switch typeU
    case 'const'
        %Piecewise constant function
        disp('number of optimization parameters n1')
        n1=n %number of optimization parameters
    case 'linear'
        %piecewise-linear function
        disp('number of optimization parameters n1')
        n1=n+1 %number of optimization parameters
    otherwise
        error('Unexpected approximation type')
end

%initial value of the optimization parameters
disp('Initial value of the optimization parameters')
b0=p(iu)*ones(1,n1)

%Interval length (all intervals are equal)
lts=(t0_te(2)-t0_te(1))/n;

%Determining lower and upper bounds for optimization parameters
disp('Lower and upper bounds for optimization parameters')
b_l=p_l(iu)*ones(1,n1)
b_u=p_u(iu)*ones(1,n1)
b=b0;
```

```
%------------------------------------------------------------

%SOLVING ODEs for initial value of the optimization parameters
nx=200;
x = linspace(t0_te(1),t0_te(2),nx);
options_ode = odeset('RelTol',1e-7,'AbsTol',1e-7);
[x,y] = ode15s(@fake_news,x,y0,options_ode);

%Plotting ODEs solutions for the initial value of optimization parameters
figure
plot1 = plot(x,y(:, 1), ...
    x,y(:, 2), ...
    x,y(:, 3));

plot1(1).LineWidth = 1.5;
plot1(2).LineWidth = 1.5;
plot1(3).LineWidth = 1.5;

plot1(1).Color = "red";
plot1(2).Color = "green";
plot1(3).Color = "blue";

plot1(1).LineStyle = "--";
plot1(2).LineStyle = "-.";
plot1(3).LineStyle = "-";

hold on
plot([t0_te(1) t0_te(2)], [y2_au y2_au],'g--')
title('SIR for b0 (initial)');
xlabel('Time t');
ylabel('Number');
legend('S','I', 'R', 'y2_au')
grid on


nsol_ode=0;
%Calculation of optimization criterion
%for the initial value of the optimization parameters
disp('Value of optimization criterion')
disp('for the initial value of the optimization parameters')
psi0_initial = fake_news_optcr(b0)

%Calculation of the value of the constraint
```

```
%for the initial value of the optimization parameters
disp('Value of constraint')
disp('for the initial value of the optimization parameters')
[constr_neq, constr_eq] = fake_news_constr(b0)

%Solving of optimization problem with bounds constrains
%on the optimization variables
disp('_______________________________________________')
disp('Solving of optimization problem with bound (dual) constrains ')
disp('on the optimization variables')

nsol_ode=0;
options_opt = optimoptions('fmincon','Algorithm', 'sqp', ...
    'MaxFunctionEvaluations', 500);
switch constraint
  case 'Yes'
    [b_opt,psio_opt] = fmincon(@fake_news_optcr,b0,[],[],[],[], ...
        b_l,b_u,@fake_news_constr, options_opt)
  case 'No'
    [b_opt,psio_opt] = fmincon(@fake_news_optcr,b0,[],[],[],[], ...
        b_l,b_u, [], options_opt)
  otherwise
    error('Unknown type of optimization problem');
end


disp('Number of solutions of ODEs (direct problem)')
nsol_ode

%Calculation of optimization criterion
%for the optimal value of the optimization parameters
disp('Value of optimization criterion')
disp('for the optimal value of the optimization parameters')
psi0_optimal = fake_news_optcr(b_opt)

%Calculation of the value of the constraint
%for the optimal value of the optimization parameters
disp('Value of constraint')
disp('for the optimal value of the optimization parameters')
[constr_neq, constr_eq] = fake_news_constr(b_opt)

%---------------------------
b=b_opt;
%-------------------------

%SOLVING ODEs for optimal value of the optimization parameters
```

```
[x,y] = ode15s(@fake_news,x,y0,options_ode);

%Plotting ODEs solutions for the optimal value of the model parameters
figure
plot2 = plot(x,y(:, 1), ...
    x,y(:, 2), ...
    x,y(:, 3));

plot2(1).LineWidth = 1.5;
plot2(2).LineWidth = 1.5;
plot2(3).LineWidth = 1.5;

plot2(1).Color = "red";
plot2(2).Color = "green";
plot2(3).Color = "blue";

plot2(1).LineStyle = "--";
plot2(2).LineStyle = "-.";
plot2(3).LineStyle = "-";

hold on
plot([t0_te(1) t0_te(2)], [y2_au y2_au],'g--')
title('SIR for b\_opt (optimal)');
xlabel('Time t');
ylabel('Number');
legend('S','I', 'R', 'y2_au')
grid on

%Plotting control function u (initial, optimal) and dual constraints on u
figure
hold on
switch typeU
    case 'const'
      t=linspace(t0_te(1),t0_te(2),n+1);
      stairs(t,[b0 b0(end)],'LineWidth',2)
      stairs(t,[b_opt b_opt(end)],'LineWidth',2)
      stairs(t,[b_l b_l(end)],'k--','LineWidth',2)
      stairs(t,[b_u b_u(end)],'k:','LineWidth',2)
      title('Control function u')
      xlabel('Time t')
      ylabel('u')
      legend('b0','b\_opt','b\_l','b\_u')
      grid on
    case 'linear'
      t=linspace(t0_te(1),t0_te(2),n1);
      plot(t,b0,'LineWidth',2)
```

```
            plot(t,b_opt,'LineWidth',2)
            plot(t,b_l,'k--','LineWidth',2)
            plot(t,b_u,'k:','LineWidth',2)
            title('Control function u');
            xlabel('Time t');
            ylabel('u');
            legend('b0','b\_opt','b\_l','b\_u')
            grid on
        otherwise
            error('Unexpected approximation type')
    end
end

function res = fake_news(t,y)
    %SIR model with global model parameters
    %
    global p

    % ----------------------------------------
    global n lts b  typeU
    global iu
    global t0_te

    nu=floor(t/lts)+1;
    nu=min(nu,n);
    switch typeU
        case 'const'
            Ut=b(nu);
        case 'linear'
            t1=t0_te(1)+(nu-1)*lts;
            t2=t1+lts;
            Ut=(t2-t)/lts*b(nu)+(t-t1)/lts*b(nu+1);
        otherwise
            error('Unexpected approximation type')
    end

    p(iu)=Ut;
    %-------------------------------------------------

    res = [
        -p(2)*y(1)*y(2)
        p(2)*y(1)*y(2) - p(1)*y(2)
        p(1)*y(2)
        ];
end
```