

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ



Оливера Лужњанин

Диференцијалне једначине и
програмски пакет *Wolfram*
Mathematica

мастер рад

Београд, 2024.

Овом приликом желим да се захвалим својој менторки, др Марији Микић, на помоћи током израде рада, времену, стрпљењу и корисним сугестијама. Такође, желела бих да изразим захвалност и осталим члановима комисије др Јелени Катичић и др Мареку Светлик на сарадњи и пренетом знању на основним и мастер студијама. Хвала мојој породици, пријатељима и колегама на подршци и разумевању.

*Београд, 2024.
Оливера Лужњанин*

Садржај

1	Увод	4
2	Диференцијалне једначине првог реда	6
2.1	Основни појмови о диференцијалним једначинама	6
2.2	Решавање диференцијалних једначина	6
2.3	Теореме о егзистенцији и јединствености решења Кошијевог проблема .	7
2.4	Геометријска интерпретација решења	8
2.5	Диференцијална једначина која раздваја променљиве	10
2.6	Линеарна диференцијална једначина првог реда	10
3	Системи обичних диференцијалних једначина првог реда	13
3.1	Основни појмови	13
3.2	Системи линеарних диференцијалних једначина првог реда	15
3.3	Динамички системи	18
4	Основне наредбе програмског пакета <i>Wolfram Mathematica</i>	20
4.1	Наредба <i>Plot</i>	21
4.2	Наредба <i>ContourPlot</i>	22
4.3	Наредба <i>DSolve</i>	23
4.4	Наредба <i>Eigenvalues</i> и <i>Eigenvectors</i>	24
4.5	Наредба <i>Wronskian</i>	26
4.6	Наредба <i>JordanDecomposition</i>	27
4.7	Наредбе <i>VectorFieldPlot</i> и <i>StreamPlot</i>	28
5	Решавање диференцијалних једначина првог реда користећи про- грамски пакет <i>Wolfram Mathematica</i>	30
5.1	Примена диференцијалних једначина	34
5.2	Њутнов закон хлађења	34
5.3	Закон радиоактивног распада	37
6	Примена система диференцијалних једначина	40
6.1	Динамички модел Предатор - плен	40
6.2	Динамички модел инфективних болести СИР	44
7	Закључак	49

Глава 1

Увод

Најразноврснији проблеми из природних наука, али и из свакодневног живота могу се моделирати диференцијалним једначинама или системима диференцијалних једначина. Отуда произилазе интересовања за проучавање метода за њихово решавање.

Диференцијалне једначине појавиле су се у време Њутна¹ и Лајбница², у седамнаестом веку, после проналаска диференцијалног и интегралног рачуна. У 18. и 19. веку откривене су многе методе за решавање диференцијалних једначина, које се и данас користе. До почетка 20. века развијено је много ефективних нумеричких метода, али је њихова примена била ограничена због ручног израчунавања. Интезивним развојем рачунара крајем 20. века, знатно се побољшава и убрзава процес решавања проблема који се могу решити нумеричким методама.

У овом раду приказано је решавање диференцијалних једначина коришћењем програмског пакета *Wolfram Mathematica*. Творац овог програмског пакета је Стивен Волфрам³. *Wolfram Mathematica* је корисна алатка за прецизно и брзо решавање одређених диференцијалних једначина. *Wolfram Mathematica* има већ уграђене наредбе помоћу којих можемо директно решити диференцијалну једначину. Такође, омогућава корисницима да једноставно и ефикасно анализирају и визуализују решења система диференцијалних једначина, што је од великог значаја у научним и инжењерским дисциплинама.

Рад је подељен на седам поглавља на следећи начин:

- Прво поглавље - је уводно и односи се на кратак увид у сва поглавља.
- Друго поглавље - односи се на обичне диференцијалне једначине првог реда. У овом делу су дефинисани основни појмови и наведене теореме о егзистенцији и јединствености решења диференцијалних једначина првог реда. У овом поглављу приказано је решавање неких типова диференцијалних једначина првог реда, као и геометријска интерпретација решења.

¹Исак Њутн (1643 — 1727), енглески физичар и математичар

²Готфрид Вилхелм Лајбниц (1646 — 1716), немачки филозоф и математичар

³Стивен Волфрам (1959, Лондон), британски физичар, математичар и програмер

- Треће поглавље - посвећено је системима диференцијалних једначина, као и основним дефиницијама и теоремама које се на њих односе. Такође овде су обрађени и нама веома значајни динамички системи.
- Четврто поглавље - бави се самим програмским пакетом *Wolfram Mathematica*. Приказане су основне наредбе овог пакета које ћемо у даљем раду користити.
- Пето поглавље - садржи примере које илуструју употребу наредби програма *Wolfram Mathematica* за диференцијалне једначине првог реда. Овде су обрађени примери Њутнов закон хлађења и закон радиоактивног распада.
- Шесто поглавље - односи се на одређене примере система диференцијалних једначина који ће бити илустровани помоћу програмског пакета *Mathematica* (динамички модел раста популације предатор - плен и динамички модел инфективних болести).
- Седмо поглавље - је закључак иза чега следи наведена литература која је коришћена при изради мастер рада.

Глава 2

Диференцијалне једначине првог реда

2.1 Основни појмови о диференцијалним једначинама

Диференцијална једначина је једначина у којој се појављује бар један од извода непознате функције. Многи физички системи се математички моделирају управо овим једначинама.

Дефиниција 1. Нека је $D \subseteq \mathbb{R}^{n+2}$ отворен скуп и нека $F : D \rightarrow \mathbb{R}$. Једначина

$$F(x, y, y', \dots, y^{(n)}) = 0,$$

где је $y = y(x)$ непозната функција и у једначини $F(x, y, y', \dots, y^{(n)}) = 0$ фигурише бар један извод непознате функције y , јесте **обична диференцијална једначина**.

Ред диференцијалне једначине одређен је редом највишег извода непознате функције који се у њој појављује.

Дакле, општи облик диференцијалне једначине првог реда ($n = 1$) је:

$$F(x, y, y') = 0,$$

а нормалан облик је:

$$y' = f(x, y),$$

где је f реална функција дефинисана на отвореном скупу $G \subseteq \mathbb{R}^2$.

2.2 Решавање диференцијалних једначина

Дефиниција 2. Функција $y = \varphi(x)$ јесте решење диференцијалне једначине $y' = f(x, y)$ на интервалу (a, b) ако је диференцијабилна на (a, b) и ако идентички задовољава једначину $y' = f(x, y)$ на (a, b) .

Дефиниција 3. *Интегрална крива* диференцијалне једначине је крива која представља график решења дате диференцијалне једначине.

Дефиниција 4. Нека су S и T отворени интервали и $S, T \subset \mathbb{R}$. Фамилија функција $y = \varphi(x, c)$, где је $c \in C \subset \mathbb{R}$ јесте **опште решење** диференцијалне једначине $y' = f(x, y)$ у области $S \times T$, ако за сваку тачку $(x_0, y_0) \in S \times T$ постоји јединствено $c_0 \in C$ такво да је $y = \varphi(x_0, c_0)$ решење диференцијалне једначине $y' = f(x, y)$ које задовољава услов $y(x_0) = y_0$.

Дефиниција 5. **Партикуларно решење** диференцијалне једначине $y' = f(x, y)$ јесте решење диференцијалне једначине $y' = f(x, y)$ које се добија из општег решења за фиксирану (допустиву) вредност параметра c .

Дефиниција 6. Решење $y = \psi(x)$ диференцијалне једначине $y' = f(x, y)$ је **сингуларно решење** ако кроз сваку његову тачку осим њега пролази и неко друго решење, које у тој тачки има исту тангенту као и решење $y = \psi(x)$, а разликује се од њега у ма којој околини тачке додира.

2.3 Теореме о егзистенцији и јединствености решења Кошијевог проблема

Дефиниција 7. **Кошијев¹ проблем (Кошијев задатак)** за диференцијалну једначину $y' = f(x, y)$ је проблем налажења решења диференцијалне једначине $y' = f(x, y)$ које задовољава услов $y(x_0) = y_0$, где су x_0, y_0 дати реални бројеви.

Нека је G отворен скуп и нека је (x_0, y_0) произвољна тачка из скупа G . Тада постоје $a, b > 0$ тако да је:

$$P = \{(x, y) | (x, y) \in \mathbb{R}^2, |x - x_0| \leq a, |y - y_0| \leq b\} \subset G. \quad (2.1)$$

Теорема 1. (Пеанова² теорема) Нека је функција f непрекидна на правоугаонику P дефинисаном једнакошћу (2.1). Тада постоји решење Кошијевог проблема $y' = f(x, y)$, $y(x_0) = y_0$ и оно је дефинисано у некој околини тачке x_0 .

Наведимо сада Липшицов услов који ћемо користити у теорему о постојању и јединствености решења Кошијевог проблема.

Дефиниција 8. Нека је $f : D \rightarrow \mathbb{R}$, $D \subset \mathbb{R}^2$. Функција f задовољава Липшицов услов на D у односу на променљиву y ако постоји константа $L > 0$ тако да за сваке две тачке $(x, y_1), (x, y_2) \in D$ важи:

$$|f(x, y_1) - f(x, y_2)| \leq L \cdot (y_1 - y_2)$$

и пишемо $f \in Lip(D, L)$.

¹Огистен Луј Коши (1789 — 1857), француски математичар

²Ђузепе Пеано (1858 — 1932), италијански математичар

Теорема 2. (Пикарова³ теорема) Нека је f непрекидна на правоугаонику

$$P = \{(x, y) | (x, y) \in \mathbb{R}^2, |x - x_0| \leq a, |y - y_0| \leq b\}$$

и нека је f Липшицова функција у односу на променљиву y на P , тј. $f \in Lip(P, L)$. Тада за сваку тачку $(x_0, y_0) \in P$ постоји јединствено решење Кошијевог проблема

$$y' = f(x, y), \quad y(x_0) = y_0$$

на интервалу $(x_0 - \alpha, x_0 + \alpha)$ где је

$$0 < \alpha < \frac{1}{L}, \quad \alpha \leq \min \left\{ a, \frac{b}{M} \right\}, \quad M = \max_{(x, y) \in P} |f(x, y)|.$$

2.4 Геометријска интерпретација решења

Решења диференцијалних једначина можемо геометријски интерпретирати.

Дефиниција 9. Нека је $y = \varphi(x)$ решење једначине

$$y' = f(x, y)$$

на неком интервалу (a, b) . Уређену тројку $(x_0, y_0, f(x_0, y_0))$ где је $f(x_0, y_0)$, y било којој тачки (x_0, y_0) , $x_0 \in (a, b)$ одређено са $y'(x_0) = f(x_0, y_0)$ називамо **линијски елемент**, а скуп свих линијских елемената је **поље правца** дате једначине.

Тангента графика решења $y = \varphi(x)$ у било којој тачки (x_0, y_0) , $x_0 \in (a, b)$ има коефицијент правца $f(x_0, y_0)$. **Изоклина** јесте геометријско место тачака у којима поље правца има исту вредност, односно сви линијски елементи имају исти нагиб.

ПРИМЕР 1. Скицирати поље правца и интегралне криве за дату диференцијалну једначину:

$$y' = y + x.$$

За дату диференцијалну једначину важи:

$$f(x, y) = y + x,$$

(јер је $y' = f(x, y)$) па једначина изоклина гласи:

$$c = y + x, \quad c \in \mathbb{R}.$$

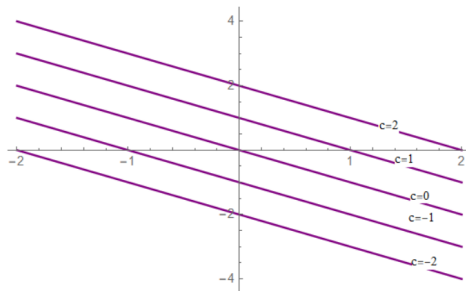
За $c = 0$, из дате једначине изоклина, добија се $y = -x$. У свакој тачки нагиб односно коефицијент правца тангенте је 0. Па је $\operatorname{tg} \alpha = 0$ (где је $\operatorname{tg} \alpha = f(x, y)$).

За $c = 1$, ће бити $y = -x + 1$ и $\operatorname{tg} \alpha = 1$

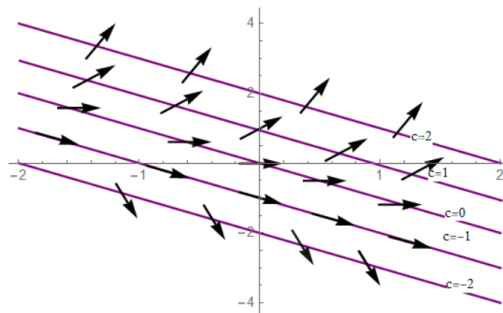
За $c = -1$, ће бити $y = -x - 1$ и $\operatorname{tg} \alpha = -1$...

³Шарл Емил Пикар (1856 – 1941), француски математичар

На слици 2.1 приказане су изоклине, док су на сликама 2.2 и 2.3 скицирана поља праваца.

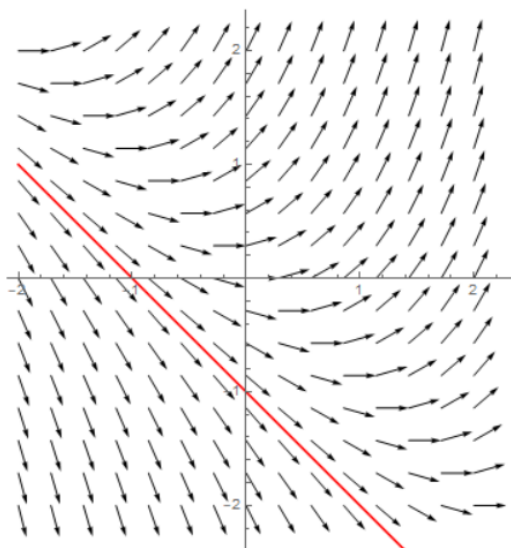


Слика 2.1: Изоклине: $y = -x + 2$, $y = -x + 1$, $y = -x$, $y = -x - 1$ и $y = -x - 2$



Слика 2.2: Поље праваца диференцијалне једначине $y' = y + x$

Поље праваца се поклопило са изоклином $y = -x - 1$ стога је $y(x) = -x - 1$ интегрална крива (слика 2.2). Та интегрална крива означена је на слици 2.3 црвеном бојом.



Слика 2.3: Интегрална крива и поље праваца за дату диференцијалну једначину $y' = y + x$

2.5 Диференцијална једначина која раздваја променљиве

Дефиниција 10. Општи облик диференцијалне једначине која раздваја променљиве је:

$$y' = f(x) \cdot g(y)$$

где је f непрекидна функција на интервалу (a, b) и g непрекидна функција на интервалу (c, d) .

По Пеановој теореме, постојаће решење Кошијевог проблема $y' = f(x) \cdot g(y)$, $y(x_0) = y_0$, за $(x_0, y_0) \in (a, b) \times (c, d)$.

Ако $g(y) \neq 0$, за свако $y \in (c, d)$, тада претходну једначину можемо записати на следећи начин:

$$\frac{y'}{g(y)} = f(x),$$

тј. после интеграције добијамо:

$$\int \frac{y' dx}{g(y)} = \int f(x) dx.$$

Увођењем смене $y(x) = t$, где је $y'(x) dx = dt$, добија се:

$$\int \frac{dt}{g(t)} = \int f(x) dx$$

односно,

$$G(t) = F(x) + c,$$

где је $G(t)$ произвољна примитивна функција функције $\frac{1}{g(t)}$ и $F(x)$ је произвољна примитивна функција функције $f(x)$, тј.

$$G(y(x)) = F(x) + c, \quad c \in \mathbb{R}.$$

Ако за неко $y_0 \in (c, d)$ важи да је $g(y_0) = 0$, онда следи да је права $y = y_0$ решење почетне диференцијалне једначине.

2.6 Линеарна диференцијална једначина првог реда

Општи облик линеарне диференцијалне једначине првог реда је:

$$a_1(x)y'(x) + a_0(x)y(x) = b(x)$$

где су $a_1(x)$, $a_0(x)$ и $b(x)$ непрекидне функције на интервалу (\tilde{a}, \tilde{b}) . Уколико $a_1(x) \neq 0$ за свако $x \in (\tilde{a}, \tilde{b})$ тада обе стране једначине можемо поделити са $a_1(x)$, при чему се

добија:

$$y' + P(x)y = Q(x), \quad (2.2)$$

где су са $P(x) = \frac{a_0(x)}{a_1(x)}$, $Q(x) = \frac{b(x)}{a_1(x)}$ означене непрекидне функције на интервалу (\tilde{a}, \tilde{b}) . Једначину (2.2) називамо **линеарна диференцијална једначина првог реда у стандардном облику**.

Уколико је $Q(x) = 0$ за свако $x \in (\tilde{a}, \tilde{b})$, онда добијамо **хомогену линеарну диференцијалну једначину првог реда**:

$$y' + P(x)y = 0. \quad (2.3)$$

Постоји неколико начина за налажење решења линеарне диференцијалне једначине првог реда. Један од начина је **метод варијације константи**. За дату једначину (2.2) решавамо одговарајућу хомогену диференцијалну једначину:

$$y' + P(x)y = 0,$$

која се своди на диференцијалну једначину која раздваја променљиве:

$$y' = -P(x)y.$$

У случају да $y \neq 0$ важи:

$$\frac{y'}{y} = -P(x)$$

односно,

$$\int \frac{y' dx}{y} = \int -P(x) dx,$$

$$\ln |y(x)| = - \int P(x) dx + c, \quad c \in \mathbb{R},$$

$$|y(x)| = e^{-\int P(x) dx} \cdot c_1, \quad c_1 > 0,$$

$$y(x) = e^{-\int P(x) dx} \cdot c_2, \quad c_2 \in \mathbb{R} \setminus \{0\}.$$

Како је и $y = 0$ решење посматране једначине, онда су сва решења једначине овог облика:

$$y(x) = e^{-\int P(x) dx} \cdot c, \quad c \in \mathbb{R}.$$

Сада варирамо константу, тј. тражимо решење нехомогене диференцијалне једначине.

Решење диференцијалне једначине (2.2) тражимо у облику:

$$y(x) = c(x) \cdot e^{-\int P(x) dx},$$

где подразумевамо да је функција $c(x)$ диференцијабилна на посматраном интервалу. Када диференцирамо добија се:

$$y'(x) = c'(x) \cdot e^{-\int P(x)dx} - c(x) \cdot e^{-\int P(x)dx} \cdot P(x).$$

Замењујемо у једначину (2.2):

$$c'(x) \cdot e^{-\int P(x)dx} - c(x) \cdot e^{-\int P(x)dx} \cdot P(x) + P(x) \cdot c(x) e^{-\int P(x)dx} = Q(x),$$

тј.

$$c'(x) \cdot e^{-\int P(x)dx} = Q(x),$$

$$c'(x) = Q(x) \cdot e^{\int P(x)dx},$$

односно,

$$c(x) = \int Q(x) \cdot e^{\int P(x)dx} dx + c.$$

Сва решења диференцијалне једначине биће овог облика:

$$y(x) = e^{-\int P(x)dx} \cdot \left(c + \int Q(x) \cdot e^{\int P(x)dx} dx \right).$$

Глава 3

Системи обичних диференцијалних једначина првог реда

3.1 Основни појмови

Нека је $D \subseteq \mathbb{R}^{2n+1}$ ($n \in \mathbb{N}$) отворен скуп у простору \mathbb{R}_2^{2n+1} и нека су $F_i : D \rightarrow \mathbb{R}$ ($1 \leq i \leq n$) произвољне функције.

Дефиниција 11. *Систем једначина облика:*

$$\begin{aligned} F_1(t, x_1(t), x_2(t), \dots, x_n(t), x'_1(t), x'_2(t), \dots, x'_n(t)) &= 0 \\ F_2(t, x_1(t), x_2(t), \dots, x_n(t), x'_1(t), x'_2(t), \dots, x'_n(t)) &= 0 \\ &\vdots \\ F_n(t, x_1(t), x_2(t), \dots, x_n(t), x'_1(t), x'_2(t), \dots, x'_n(t)) &= 0, \end{aligned} \tag{3.1}$$

где је t независна променљива, $x_i = x_i(t)$ непознате функције дефинисане на неком интервалу $I \subseteq \mathbb{R}$, а $x'_i = x'_i(t)$ изводи функција x_i назива се **систем обичних диференцијалних једначина првог реда у општем облику**.

Систем диференцијалних једначина који има облик:

$$\begin{aligned} x'_1(t) &= f_1(t, x_1(t), x_2(t), \dots, x_n(t)) \\ x'_2(t) &= f_2(t, x_1(t), x_2(t), \dots, x_n(t)) \\ &\vdots \\ x'_n(t) &= f_n(t, x_1(t), x_2(t), \dots, x_n(t)), \end{aligned} \tag{3.2}$$

где су реалне функције $f_i \in C(G)$ ($1 \leq i \leq n$) дефинисане на области $G \subseteq \mathbb{R}^{n+1}$, назива се **нормални систем обичних диференцијалних једначина првог реда**. Систем (3.2) можемо приказати у векторском облику, тиме његово проучавање постаје аналогно проучавању једне обичне диференцијалне једначине. Са $X(t)$ означимо вектор $(x_1(t), x_2(t), \dots, x_n(t))$, а са $X'(t)$ вектор колону извода $(x'_1(t), x'_2(t), \dots, x'_n(t))$, онда се овај систем може записати у облику:

$$X' = F(t, X), \tag{3.3}$$

где је $F(t, X) = (f_1(t, x_1(t), x_2(t), \dots, x_n(t)), \dots, f_n(t, x_1(t), x_2(t), \dots, x_n(t)))$. Уређена n -торка реалних непрекидно диференцијабилних функција $(\tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_n(t))$ на интервалу $I \subset \mathbb{R}$ зове се **решење** нормалног система обичних диференцијалних једначина (3.2) ако за свако $t \in I$ важе једнакости:

$$\begin{aligned}\tilde{x}'_1(t) &= f_1(t, \tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_n(t)) \\ \tilde{x}'_2(t) &= f_2(t, \tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_n(t)) \\ &\vdots \\ \tilde{x}'_n(t) &= f_n(t, \tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_n(t)).\end{aligned}$$

Скуп $\{(t, \tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_n(t)) \in \mathbb{R}^{n+1} \mid t \in I\} \subset G$ који графички представља решење система (3.2) назива се **интегрална крива** тог система.

Дефинишимо сада појам општег решења система диференцијалних једначина.

Дефиниција 12. Функције $x_i = x_i(t, c_1, \dots, c_n)$, за $1 \leq i \leq n$, где су c_1, \dots, c_n параметри и $t \in I$, називамо **опште решење** система (3.2) на отвореном интервалу $\tilde{G} = I \times J \subset \mathbb{R}^{n+1}$ ако за произвољну тачку $(t_0, x_1^{(0)}, \dots, x_n^{(0)}) \in \tilde{G}$ систем једначина $x_i^{(0)} = x_i(t_0, c_1, \dots, c_n)$ има јединствено решење $(c_1^{(0)}, \dots, c_n^{(0)})$ тако да су функције $x_i = x_i(t, c_1^{(0)}, \dots, c_n^{(0)})$ решење система (3.2) које задовољавају услов $x_i(t_0, c_1^{(0)}, \dots, c_n^{(0)}) = x_i^{(0)}$.

Партикуларно решење система (3.2) добија се из општег решења система фиксирањем константи c_i , $1 \leq i \leq n$. Углавном се партикуларна решења система издвајају из општег решења система диференцијалних једначина, јер се захтева да испуњавају неке посебне услове, као што је Кошијев проблем.

Нека је $(t_0, x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \in G$ фиксирана тачка. **Кошијев проблем** може се формулисати на следећи начин: Одредити оно решење $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ система (3.2) које задовољава почетне услове:

$$\tilde{x}_1(t_0) = x_1^{(0)}, \tilde{x}_2(t_0) = x_2^{(0)}, \dots, \tilde{x}_n(t_0) = x_n^{(0)}.$$

Односно, треба одредити интегралну криву система (3.2) која пролази кроз тачку

$$(t_0, x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}).$$

Теорема 3. Нека је $F : G \rightarrow \mathbb{R}^n$ непрекидна функција на отвореном скупу $G \subseteq \mathbb{R}^{n+1}$ и нека је $(t_0, X_0) \in G$ фиксирана тачка. Тада непрекидно диференцијабилна функција $X : (a, b) \rightarrow \mathbb{R}^n$ јесте решење Кошијевог проблема $X' = F(t, X(t))$, $X(t_0) = X_0$ ако и само ако је X непрекидно решење интегралне једначине:

$$X(t) = X_0 + \int_{t_0}^t F(s, X(s)) ds, \quad (3.4)$$

за свако $t \in (a, b)$.

Теорема 4. (Пеанова теорема) Нека је $F : G \rightarrow \mathbb{R}^n$ непрекидна функција на отвореном скупу $G \subset \mathbb{R}^{n+1}$. Тада за сваку тачку $(t_0, X_0) \in G$ постоји бар једно

решење Кошијевог проблема $X' = F(t, X(t))$, $X(t_0) = X_0$ дефинисано на отвореном интервалу (a, b) које садржи тачку t_0 .

Теорема 5. (Пикарова теорема) Нека је $F : G \rightarrow \mathbb{R}^n$ непрекидна функција и локално Липшицова на отвореном скупу $G \subset \mathbb{R}^{n+1}$ у односу на променљиву X . Тада за свако $(t_0, X_0) \in G$ постоји јединствено решење Кошијевог проблема $X' = F(t, X(t))$, $X(t_0) = X_0$ дефинисано на отвореном интервалу $(a, b) \subseteq \mathbb{R}$ које садржи тачку t_0 .

3.2 Системи линеарних диференцијалних једначина првог реда

Нормални систем линеарних диференцијалних једначина првог реда има облик:

$$\begin{aligned} x'_1(t) &= a_{11}(t)x_1(t) + a_{12}(t)x_2(t) + \dots + a_{1n}(t)x_n(t) + b_1(t) \\ x'_2(t) &= a_{21}(t)x_1(t) + a_{22}(t)x_2(t) + \dots + a_{2n}(t)x_n(t) + b_2(t) \\ &\vdots \\ x'_n(t) &= a_{n1}(t)x_1(t) + a_{n2}(t)x_2(t) + \dots + a_{nn}(t)x_n(t) + b_n(t), \end{aligned}$$

где су са $a_{ij} : (a, b) \rightarrow \mathbb{R}$ и $b_i : (a, b) \rightarrow \mathbb{R}$ ($1 \leq i, j \leq n$ и $(a, b) \subseteq \mathbb{R}$) задате непрекидне функције. Овај систем може се записати у следећем облику:

$$X'(t) = A(t) \cdot X(t) + B(t), \quad (3.5)$$

где су A , X , и B матрице:

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_n(t) \end{bmatrix} \quad \text{и} \quad X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}.$$

Функција $F : G \rightarrow \mathbb{R}^n$, где је $G = (a, b) \times \mathbb{R}^n$, облика:

$$F(t, X) = A(t) \cdot X(t) + B(t)$$

је непрекидна и локално Липшицова на (a, b) , па из Пикарове теореме следи да за сваку тачку $(t_0, X_0) \in (a, b) \times \mathbb{R}^n$ Кошијев проблем:

$$X'(t) = A(t) \cdot X(t) + B(t), \quad X(t_0) = X_0 \quad (3.6)$$

има јединствено решење $\tilde{X} = \tilde{X}(t)$ дефинисано на неком интервалу $(\tilde{a}, \tilde{b}) \subseteq (a, b)$. Може се показати да је $(\tilde{a}, \tilde{b}) = (a, b)$.

Хомоген систем диференцијалних једначина има облик:

$$X'(t) = A(t) \cdot X(t) \quad (3.7)$$

Нека је $L_n(X)(t) = X'(t) - A(t)X(t)$ линеарни диференцијални оператор, онда се претходни систем може записати у облику једначине:

$$L_n(X)(t) = 0_{n \times 1}.$$

Теорема 6. Ако су функције $X_k = X_k(t)$ ($1 \leq k \leq n$) решења система (3.7), онда је и функција

$$X(t) = \sum_{k=1}^n c_k \cdot X_k(t)$$

такође решење система (3.7), за свако $(c_1, c_2, \dots, c_n \in \mathbb{R})$.

Доказ: $X_k = X_k(t)$ су решења система па важи $L_n(X_k)(t) = 0$, за $1 \leq k \leq n$. Стога следи,

$$L_n(X)(t) = L_n\left(\sum_{k=1}^n c_k \cdot X_k\right)(t) = \sum_{k=1}^n c_k \cdot L_n(X_k)(t) = 0. \quad \square$$

Дефиниција 13. Векторске функције $X_i(t) = \begin{bmatrix} x_{1i}(t) \\ x_{2i}(t) \\ \vdots \\ x_{ni}(t) \end{bmatrix}$, $1 \leq i \leq n$ су **линеарно**

независне на $(a, b) \subseteq \mathbb{R}$ ако за константе $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{R}$ из $\alpha_1 X_1(t) + \alpha_2 X_2(t) + \dots + \alpha_n X_n(t) = 0_{n \times 1}$, за свако $t \in (a, b)$ следи

$$\alpha_1 = \alpha_2 = \dots = \alpha_n = 0.$$

У супротном су линеарно зависне.

Дефиниција 14. Нека су $X_i(t) = \begin{bmatrix} x_{1i}(t) \\ x_{2i}(t) \\ \vdots \\ x_{ni}(t) \end{bmatrix}$, $1 \leq i \leq n$, векторске функције. Тада се

детерминанта

$$W(X_1, X_2, \dots, X_n)(t) = \begin{vmatrix} x_{11}(t) & x_{12}(t) & \dots & x_{1n}(t) \\ x_{21}(t) & x_{22}(t) & \dots & x_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}(t) & x_{n2}(t) & \dots & x_{nn}(t) \end{vmatrix}$$

назива детерминанта **Вронског**¹ (**Вронскијан**) система $\{X_i\}$.

Теорема 7. Ако су функције $X_i = X_i(t)$ ($1 \leq i \leq n$) линеарно зависне на $(a, b) \subset \mathbb{R}$, онда је $W(X_1, X_2, \dots, X_n)(t) = 0$, за свако $t \in (a, b)$.

¹Јозеф Хоене-Вронски (1778 — 1853), пољски филозоф, математичар и физичар

Теорема 8. Нека су функције $a_{ij} = a_{ij}(t)$ непрекидне на $(a, b) \subseteq \mathbb{R}$. Ако функције $X_i = X_i(t)$ ($1 \leq i \leq n$) јесу линеарно независна решења система (3.7) на (a, b) , онда за свако $t \in (a, b)$ важи

$$W(X_1, \dots, X_n)(t) \neq 0.$$

Последица 1. Нека су $a_{ij} \in C(a, b)$. Тада решења $X_i = X_i(t)$ ($1 \leq i \leq n$) система (3.7) јесу линеарно независна на (a, b) ако и само ако постоји $t_0 \in (a, b)$ такво да важи

$$W(X_1, \dots, X_n)(t_0) \neq 0.$$

Дефиниција 15. Нека су

$$X_i(t) = \begin{bmatrix} x_{1i}(t) \\ x_{2i}(t) \\ \vdots \\ x_{ni}(t) \end{bmatrix}, 1 \leq i \leq n$$

линеарно независна решења система (3.7) на (a, b) . Скуп који се састоји од n линеарно независних решења назива се **фундаментални скуп** или **фундаментални систем** решења тог система.

Матрица

$$M(t) = \begin{bmatrix} x_{11}(t) & x_{12}(t) & \dots & x_{1n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}(t) & x_{n2}(t) & \dots & x_{nn}(t) \end{bmatrix}$$

је **фундаментална матрица** система (3.7) на интервалу (a, b) .

Видимо да из теореме 8 следи да за свако $t \in (a, b)$

$$\det M(t) = W(X_1, \dots, X_n)(t) \neq 0.$$

Теорема 9. Ако је $M(t)$ фундаментална матрица система (3.7) на интервалу $(a, b) \subseteq \mathbb{R}$ онда је:

$$X(t) = M(t) \cdot C, \text{ где је } C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}, (c_i \in \mathbb{R})$$

опште решење тог система у области $(a, b) \times \mathbb{R}^n$.

Нехомоген систем

$$X'(t) = A(t)X(t) + B(t) \tag{3.8}$$

уз помоћ линеарног диференцијалног оператора L_n може се записати у облику:

$$L_n(x)(t) = B(t)$$

Следећа теорема даје нам опште решење нехомогеног система.

Теорема 10. Нека су $a_{ij}, b_i \in C(a, b)$ ($1 \leq i, j \leq n$), где је $(a, b) \subseteq \mathbb{R}$. Ако је $X_{pnh} = X_{pnh}(t)$ произвољно решење нехомогеног система (3.8), а $X_{oh}(t) = \sum_{i=1}^n c_i X(t)$ ($c_i \in \mathbb{R}$) опште решење хомогеног система (3.7) у области $(a, b) \times \mathbb{R}^n$, онда је формулом

$$X(t) = X_{oh}(t) + X_{pnh}(t)$$

дато опште решење система (3.8) у области $(a, b) \times \mathbb{R}^n$.

Хомогени систем диференцијалних једначина са константним коефицијентима је облика:

$$X'(t) = A \cdot X(t), \quad (3.9)$$

где је $A = (a_{ij})_{n \times n}$ константна квадратна матрица, $a_{ij} \in \mathbb{R}$ ($1 \leq i, j \leq n$).

Матрица $M(t) = e^{At}$ је фундаментална матрица система (3.9) за $t \in (a, b)$.

Теорема 11. Опште решење система (3.9) у области $\mathbb{R} \times \mathbb{R}^n$ је:

$$X(t) = e^{tA} \cdot C, \quad C \in \mathbb{R}^n,$$

за свако $t \in \mathbb{R}$.

Кошијево решење $\tilde{X} = \tilde{X}(t)$ система (3.9) које задовољава почетни услов $X(t_0) = X_0$ је облика:

$$\tilde{X}(t) = e^{(t-t_0)A} \cdot X_0.$$

3.3 Динамички системи

Дефиниција 16. Динамички систем

$$\begin{aligned} x'_1(t) &= f_1(x_1(t), \dots, x_n(t)) \\ x'_2(t) &= f_2(x_1(t), \dots, x_n(t)) \\ &\vdots \\ x'_n(t) &= f_n(x_1(t), \dots, x_n(t)), \end{aligned} \quad (3.10)$$

где су $x_1(t), \dots, x_n(t)$ променљиве система, t време и $f_1(x_1(t), \dots, x_n(t)), f_2(x_1(t), \dots, x_n(t)), \dots, f_n(x_1(t), \dots, x_n(t))$ функције које описују како се променљиве система мењају током времена, је систем диференцијалних једначина у нормалном облику у ком се не појављује независна променљива.

Динамички систем (3.10) можемо и једноставније записати:

$$X' = F(X), \quad (3.11)$$

при чему је $X = (x_1(t), \dots, x_n(t))$, а $F = (f_1, f_2, \dots, f_n)$.

Дефиниција 17. Фазне трајекторије динамичког система су пројекције интегралних кривих на фазни простор паралелно t -оси.

Фазни портрет динамичког система чине графици фазних трајекторија са назначеним смеровима фазних трајекторија.

Дефиниција 18. Тачка X^* је положај равнотеже (еквистријум) динамичког система (3.11), ако је $F(X^*) = 0$.

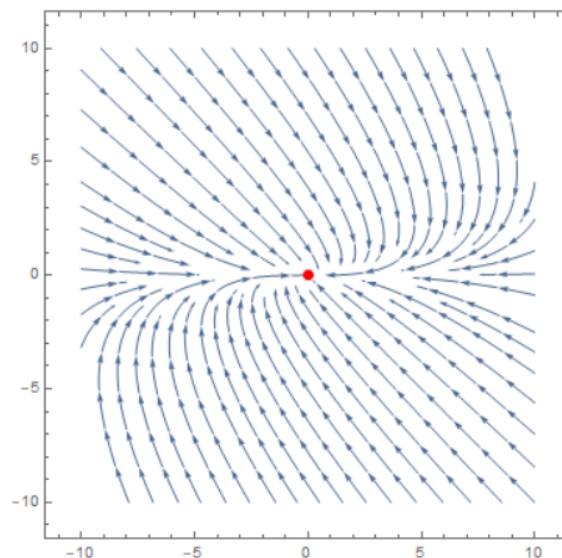
Фазне трајекторије могу бити:

- тачке (еквистријуми),
- глатке криве без самопресека,
- затворене глатке криве.

ПРИМЕР 2. За дати динамички систем наћи еквистријум и приказати фазне трајекторије:

$$\begin{aligned}x' &= -x + 2y \\y' &= -3y.\end{aligned}\tag{3.12}$$

Из дефиниције 18, нека тачка (x_e, y_e) је еквистријум система (3.12) ако важи $-x_e + 2y_e = 0$ и $-3y_e = 0$. Решавањем овог тривијалног система добија се да је еквистријум тачка $(0, 0)$. На слици 3.1 приказане су фазне трајекторије са назначеним смеровима тј. фазни портрет, а еквистријум је истакнут црвеном бојом.



Слика 3.1: Фазни портрет динамичког система (3.12)

Глава 4

Основне наредбе програмског пакета *Wolfram Mathematica*

Предности употребе програмског пакета *Wolfram Mathematica* у проучавању диференцијалних једначина су бројне, а можда је и најкориснија могућност цртање графика функција и визуално представљање решења диференцијалних једначина и система диференцијалних једначина.

Неке од основних наредби са којима ћемо се упознати и које ћемо користити у даљем раду су:

- Наредба *Plot* - користи се за цртање графика функције једне променљиве.
- Наредба *ContourPlot* - ова наредба скицира контуре у којима функција има исту вредност или график имплицитно задате функције.
- Наредба *DSolve* - користи се за решавање диференцијалне једначине или система диференцијалних једначина;
- Наредба *Eigenvalues* и *Eigenvectors* - помоћу ових наредби израчунавају се сопствене вредности и сопствени вектори задате матрице. У контексту диференцијалних једначина, ове наредбе користиће се за анализу стабилности и испитивање понашања решења система диференцијалних једначина.
- Наредба *Wroskian* - израчунава детерминанту Вронског. Важан је алат за испитивање линеарне независности решења диференцијалних једначина.
- Наредба *JordanDecomposition* - ова наредба, за задату квадратну матрицу, рачуна и исписује матрицу сличности и Жорданов канонски облик дате квадратне матрице. Веома је корисна наредба за динамичке системе чија матрица има вишеструке сопствене вредности.
- Наредба *VectorFieldPlot* и *StreamPlot* - ове наредбе су важне за визуелно представљање векторског поља диференцијалних једначина испртавањем поља праваца и за приказивање фазних трајекторија.

4.1 Наредба *Plot*

Једна од основних наредби у програмском пакету *Wolfram Mathematica* је наредба *Plot*. Наредба *Plot* исцртава график произвољне функције једне променљиве на задатом интервалу. Наредба у програмском пакету *Wolfram Mathematica* уз помоћ које цртамо график функције $f = f(x)$ на сегменту $[x_{min}, x_{max}]$ је:

$$Plot[f, \{x, x_{min}, x_{max}\}].$$

Такође, могуће је и цртање графика више функција у истом координатном систему, и то се постиже наредбом:

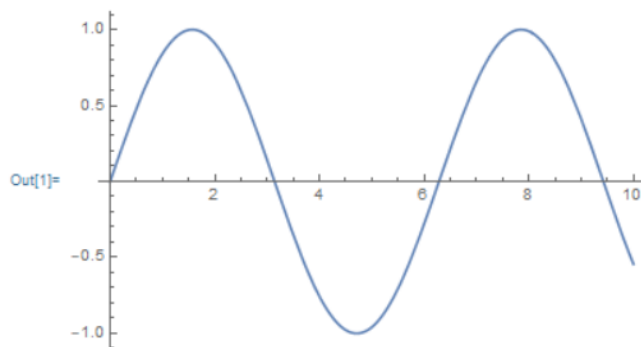
$$Plot[\{f_1, f_2, f_3, \dots, f_n\}, \{x, x_{min}, x_{max}\}],$$

где су са $f_1, f_2, f_3, \dots, f_n$ означене функције променљиве x на истом сегменту $[x_{min}, x_{max}]$.

ПРИМЕР 3. У програмском пакету *Wolfram Mathematica* нацртати график функције $f(x) = \sin x$ на сегменту $[0, 10]$.

У линији *In[n]* корисник уноси своју наредбу, а *Wolfram Mathematica* враћа резултат у линији *Out[n]*, где је n природни број који служи за бројење унетих наредби. Решење је приказано на слици 4.1.

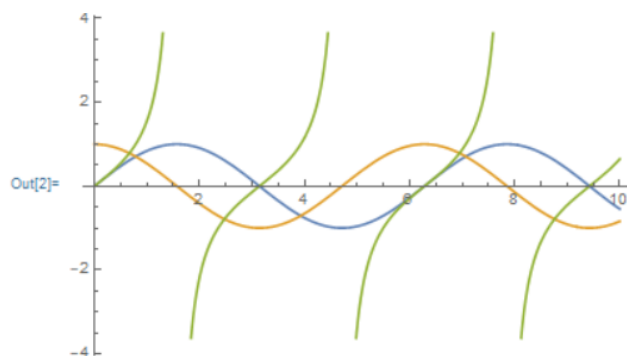
```
In[1] := Plot[Sin[x], {x, 0, 10}]
```



Слика 4.1: График функције $f(x) = \sin x$ скициран помоћу наредбе *Plot*

ПРИМЕР 4. Помоћу програмског пакета *Wolfram Mathematica*, скицирати графике функција $f_1(x) = \sin x$, $f_2(x) = \cos x$, $f_3(x) = \operatorname{tg} x$ на сегменту $[0, 10]$.

```
In[2] := Plot[{Sin[x], Cos[x], Tan[x]}, {x, 0, 10}]
```



Слика 4.2: Графици функција $f_1(x) = \sin x$, $f_2(x) = \cos x$, $f_3(x) = \operatorname{tg} x$ нацртани помоћу наредбе *Plot*

4.2 Наредба *ContourPlot*

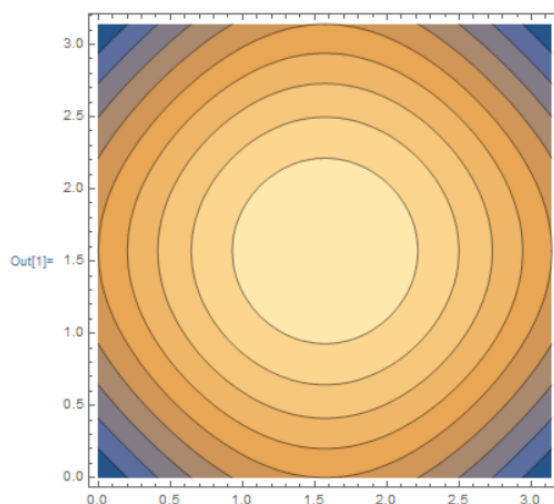
Наредба *ContourPlot* скицира контуре у којима функција има исту вредност или график имплицитно задате функције једне променљиве. Наредба која се задаје у програмском пакету *Wolfram Mathematica* је:

$$\text{ContourPlot}[f, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}],$$

где је са f означена функција $f = f(x, y)$, где је $x \in [x_{\min}, x_{\max}]$ и $y \in [y_{\min}, y_{\max}]$.

ПРИМЕР 5. Помоћу програмског пакета *Wolfram Mathematica*, скицирати контуре дате функције $f(x, y) = \sin x + \sin y$, где је $x \in [0, \pi]$ и $y \in [0, \pi]$.

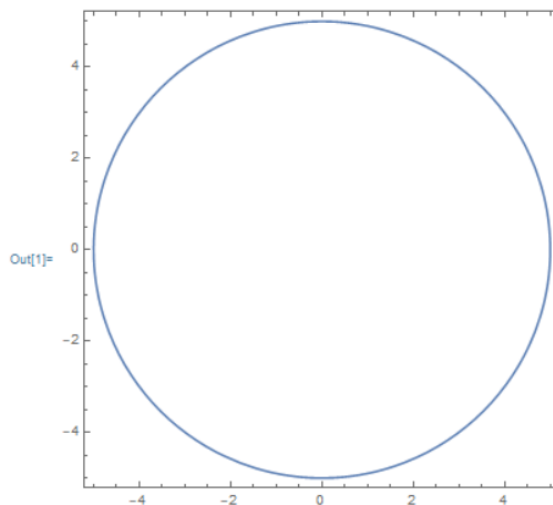
```
In[1] := ContourPlot[Sin[x] + Sin[y], {x, 0, Pi}, {y, 0, Pi}]
```



Слика 4.3: Контуре функције $f(x, y) = \sin x + \sin y$, скициране помоћу наредбе *ContourPlot*

ПРИМЕР 6. Помоћу програмског пакета *Wolfram Mathematica*, скицирати криву $x^2 + y^2 = 25$, где је $x \in [-5, 5]$ и $y \in [-5, 5]$.

```
In[1] := ContourPlot[x^2 + y^2 == 25, {x, -5, 5}, {y, -5, 5}]
```



Слика 4.4: Графици имплицитно задатих функција које су задате једначином $x^2 + y^2 = 25$, скициран помоћу наредбе *ContourPlot*

4.3 Наредба *DSolve*

Наредба *DSolve* може да решава диференцијалне једначине или систем диференцијалних једначина аналитички, за оне једначине за које је могуће аналитички одредити решење. Наредба у програму *Wolfram Mathematica* уз помоћ које долазимо до решења диференцијалне једначине је:

$$DSolve[eqn, y[x], x],$$

где је са *eqn* означена диференцијална једначина, $y[x]$ је непозната функција, а x је независна променљива.

Позивањем наредбе:

$$DSolve[\{eqn1, eqn2, \dots, eqnn\}, \{y_1, y_2, \dots, y_n\}, x]$$

решава се систем диференцијалних једначина за непознате функције $y_1 = y_1(x), \dots, y_n = y_n(x)$, где је x независна променљива.

На слици 4.5 приказано је опште решења диференцијалне једначине $y'(x) = ay(x)$, где је a реални параметар, добијено коришћењем наредбе *DSolve*.

```
In[1] := DSolve[y'[x] == a * y[x], y[x], x]
Out[1] = {{y[x] -> e^{ax} c_1}}
```

Слика 4.5: Опште решење диференцијалне једначине $y' = ay(x)$ добијено наредбом *DSolve*

Наредбу *DSolve* можемо користити и за одређивање решења Кошијевог проблема. Наредба се задаје на следећи начин:

$$DSolve[\{eqn, y[a] == b\}, y[x], x],$$

где је са *eqn* означена диференцијална једначина чије решење треба да задовољава услов $y(a) = b$.

Пример како решавамо Кошијев проблем аналитички и помоћу програма *Wolfram Mathematica* биће приказан у глави 5.

4.4 Наредба *Eigenvalues* и *Eigenvectors*

Wolfram Mathematica поседује наредбе за одређивање сопствених вредности и сопствених вектора матрица.

Сопствене вредности матрице $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ представљају корене карактеристичног полинома:

$$\det(M - \lambda E) = \begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = (a - \lambda)(d - \lambda) - bc = 0,$$

тј.

$$\lambda^2 - (a + d)\lambda + ad - bc = 0.$$

Теорема 12. *Ако су сопствене вредности матрице реалне и различите онда су сопствени вектори линеарно независни.*¹

Дакле, уколико су корени карактеристичног полинома λ_1, λ_2 реални и различити, тада сопственим вредностима одговарају не-нула линеарно независни сопствени вектори γ_1 и γ_2 које добијамо решавањем једначина: $M\gamma_1 = \lambda_1\gamma_1$ и $M\gamma_2 = \lambda_2\gamma_2$.

У програмском пакету *Wolfram Mathematica*, *Eigenvalues[M]* је наредба која нам даје све сопствене вредности матрице M , док наредбом *Eigenvectors[M]* добијамо све сопствене векторе матрице M .

ПРИМЕР 7. *Нека је дата матрица*

$$M = \begin{bmatrix} -1 & 3 \\ 5 & -3 \end{bmatrix}.$$

Користићењем програмског пакета Wolfram Mathematica, наћи сопствене вредности и сопствене векторе матрице M.

¹Значење линеарно независни је објашњено у дефиницији 13, у поглављу 3.2

```
In[1] := M = {{-1, 3}, {5, -3}}
Out[1] = {{-1, 3}, {5, -3}}
In[2] := Eigenvalues[M]
Out[2] = {-6, 2}
In[3] := Eigenvectors[M]
Out[3] = {{-3, 5}, {1, 1}}
```

Слика 4.6: Сопствене вредности и сопствени вектори матрице M добијене помоћу наредби *Eigenvalues* и *Eigenvectors*

У општем случају, корени карактеристичног полинома могу бити и једнаки. Вишеструкост броја λ као решење карактеристичне једначине је алгебарска вишеструкост. Геометријска вишеструкост сопствене вредности је димензија њеног карактеристичног потпростора тј. геометријска вишеструкост се може објаснити као број линеарно независних карактеристичних вектора који одговарају истом карактеристичном корену λ . Алгебарска вишеструкост увек је већа или једнака геометријској вишеструкости.

Нека је сада дата матрица:

$$M_1 = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{bmatrix}.$$

Ова матрица има један карактеристични корен $\lambda = 2$, чија је алгебарска вишеструкост једнака 3. Међутим постоји само један линеарно независан сопствени вектор γ_1 који одговара сопственој вредности $\lambda = 2$, па је геометријска вишеструкост једнака 1. У овом случају, када у програмском пакету *Wolfram Mathematica* позовемо наредбе *Eigenvalues* и *Eigenvectors*, за сопствене векторе нам враћа један сопствени вектор и друга два представљена као нула вектори (слика 4.7). С обзиром на то да је сопствени вектор не-нула вектор, *Wolfram Mathematica* нам не даје добро решење. Може се искористити наредба *JordanDecomposition*. Ова наредба може пружити тачне уопштене сопствене векторе и допунити информације које се не могу директно добити из наредби *Eigenvalues* и *Eigenvectors*. Више о овој наредби погледати у поглављу 4.6.

```
In[1] := M1 = {{2, 1, 0}, {0, 2, 1}, {0, 0, 2}}
Out[1] = {{2, 1, 0}, {0, 2, 1}, {0, 0, 2}}
In[2] := Eigenvalues[M1]
Out[2] = {2, 2, 2}
In[3] := Eigenvectors[M1]
Out[3] = {{1, 0, 0}, {0, 0, 0}, {0, 0, 0}}
```

Слика 4.7: Сопствене вредности и сопствени вектори матрице M_1 добијене помоћу наредби *Eigenvalues* и *Eigenvectors* када алгебарска вишеструкост није једнака геометријској вишеструкости

4.5 Наредба *Wronskian*

Уколико уз помоћ Вронскијана покажемо линеарну независност функција, то ће нам бити од кључног значаја за одређивање базних решења линеарних диференцијалних једначина и система диференцијалних једначина.

У програмском пакету *Wolfram Mathematica* за израчунавање детерминанте Вронског, користимо наредбу *Wronskian*. Запис ове наредбе је:

$$\text{Wronskian}[\{f_1, f_2, \dots, f_n\}, x],$$

где су f_1, f_2, \dots, f_n функције које зависе од x .

Постоји још неколико начина за позивање ове наредбе:

- $\text{Wronskian}[eqn, y, x]$ - Вронскијан за решења линеарне диференцијалне једначине, где је са eqn означена линеарна диференцијална једначина непознате функције $y(x)$, а x је независна променљива;
- $\text{Wronskian}[\{eqns\}, \{y_1, y_2, \dots, y_n\}, x]$ - Вронскијан за решења система линеарних диференцијалних једначина $eqns$.

На пример, нека су дате функције $y_1(x) = e^x$, $y_2(x) = e^{3x}$. На слици 4.8 одређен је Вронскијан ових функција коришћењем наредбе *Wronskian*.

```
In[1] := Wronskian[{Exp[x], Exp[3x]}, x]
Out[1] = 2e4x
```

Слика 4.8: Вронскијан функција $y_1(x) = e^x$, $y_2(x) = e^{3x}$ добијен помоћу наредбе *Wronskian*

ПРИМЕР 8. *Одредити Вронскијан дате линеарне диференцијалне једначине $y'(x) = y(x)$.*

```
In[2] := Wronskian[{y'[x] == y[x]}, y, x]
Out[2] = ex
```

Слика 4.9: Вронскијан линеарне диференцијалне једначине $y'(x) = y(x)$ добијен помоћу наредбе *Wronskian*

ПРИМЕР 9. *Одредити Вронскијан за дати систем диференцијалних једначина*

$$\begin{aligned} y_1' &= 3y_1 + 4y_2 \\ y_2' &= -4y_1 + 3y_2. \end{aligned} \tag{4.1}$$

Такође и у овом примеру, можемо директно без налажења решења система диференцијалних једначина израчунати Вронскијан.

```
In[3] := Wronskian[{y1'[x] == 3y1[x] + 4y2[x], y2'[x] == -4y1[x] + 3y2[x]}, {y1, y2}, x]
Out[3] = e6x
```

Слика 4.10: Вронскијан за дати систем диференцијалних једначина (4.1) добијен помоћу наредбе *Wronskian*

4.6 Наредба *JordanDecomposition*

Једна од уграђених наредби у програмском пакету *Wolfram Mathematica* је наредба *JordanDecomposition*. За дату квадратну матрицу M , наредба нам враћа листу која се састоји од $\{S, J\}$, где је S матрица сличности, а J Жорданов канонски облик матрице M .

Дефиниција 19. Матрице M и S су сличне ако постоји регуларна матрица P тако да је $S = P^{-1}MP$.

ПРИМЕР 10. За дату матрицу M користећи наредбу *JordanDecomposition* одредити матрицу сличности и Жорданов канонски облик матрице M :

$$M = \begin{bmatrix} 81 & 25 & 0 \\ 0 & 121 & 0 \\ 9 & 6 & 3 \end{bmatrix}$$

```
In[1] := JordanDecomposition[{{81, 25, 0}, {0, 121, 0}, {9, 6, 3}}]
Out[1] = {{{0, 26, 590}, {0, 0, 944}, {1, 3, 93}}, {{3, 0, 0}, {0, 81, 0}, {0, 0, 121}}}
```

Слика 4.11: Матрица сличности и Жорданов канонски облик матрице M добијене наредбом *JordanDecomposition*

На основу Жордановог облика дате матрице M , можемо закључити да су сопствене вредности ове матрице реалне и различите. Ово тврђење можемо проверити користећи наредбе *Eigenvalues* и *Eigenvectors*. Најпре, унесимо у програм све елементе матрице M , а затим позовимо претходно поменуто наредбе:

```
In[2] := M = {{81, 25, 0}, {0, 121, 0}, {9, 6, 3}};
In[3] := Eigenvalues[M]
Out[3] = {121, 81, 3}
In[4] := Eigenvectors[M]
Out[4] = {{590, 944, 93}, {26, 0, 3}, {0, 0, 1}}
```

Слика 4.12: Сопствене вредности и сопствени вектори матрице M

На слици 4.12 уочавамо да су сопствене вредности $\lambda_1 = 121$, $\lambda_2 = 81$ и $\lambda_3 = 3$, што потврђује претходно добијени резултат.

4.7 Наредбе *VectorFieldPlot* и *StreamPlot*

У програмском пакету *Wolfram Mathematica* постоје наредбе помоћу којих можемо графички представити решења без решавања диференцијалне једначине, односно система диференцијалних једначина. Понекад није могуће решити диференцијалну једначину те нам је корисно имати неку представу о графику решења или о понашању решења система диференцијалних једначина.

Функција *VectorFieldPlot* нам даје могућност да визуелно представимо векторско поље диференцијалних једначина исцртавањем поља праваца.

Запис наредбе *VectorFieldPlot* је:

$$\text{VectorFieldPlot}[\{f(x, y), g(x, y)\}, \{x, x_{\min}, x_{\max}\}, \{y, y_{\min}, y_{\max}\}],$$

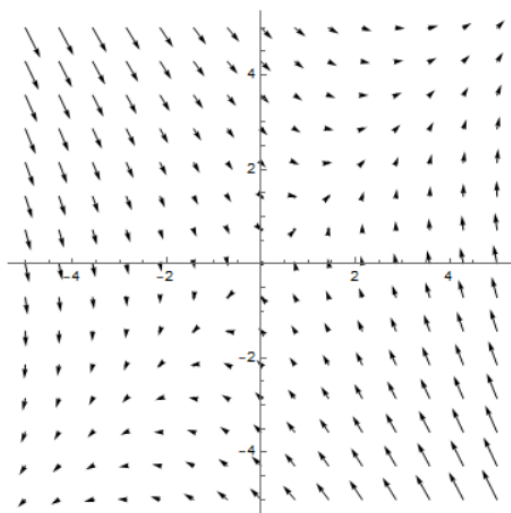
где је са $f(x, y)$ и $g(x, y)$ обележен систем диференцијалних једначина $\begin{cases} x' = f(x, y) \\ y' = g(x, y) \end{cases}$ за $x \in (x_{\min}, x_{\max}), y \in (y_{\min}, y_{\max})$.

ПРИМЕР 11. *Посматрајмо следећи систем диференцијалних једначина:*

$$\begin{aligned} x' &= -x + 3y \\ y' &= 5x - 3y \end{aligned} \tag{4.2}$$

Користећи наредбу VectorFieldPlot^2 исцртати поље праваца датог система диференцијалних једначина (4.2):

```
In[2] := VectorFieldPlot[{-x + 3y, 5x - 3y}, {x, -5, 5}, {y, -5, 5}]
Out[2] =
```



Слика 4.13: Поље праваца система диференцијалних једначина (4.2) добијено коришћењем наредбе *VectorFieldPlot*

²У програмском пакету *Wolfram Mathematica* да бисмо могли користити ову наредбу потребно је укључити пакет `Graphics` и позвати наредбу `Needs["VectorFieldPlots`"]`.

Систем диференцијалних једначина који смо поменули у претходном примеру јесте **динамички систем**.

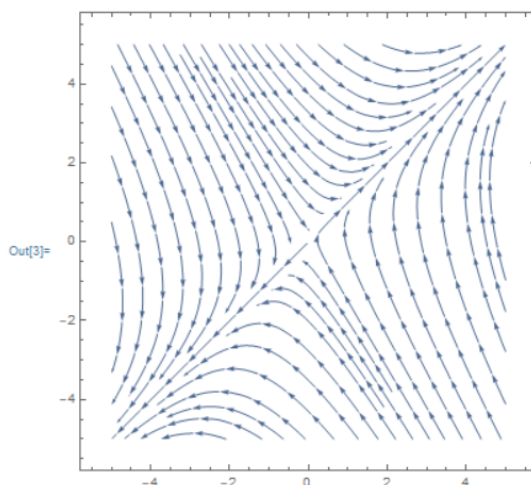
Фазни портрет динамичког система исцртава се наредбом *StreamPlot*:

$$\text{StreamPlot}\{f(x, y), g(x, y)\}, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\},$$

где су са $f(x, y)$ и $g(x, y)$ обележене функције које фигуришу у динамичком систему $\begin{cases} x' = f(x, y) \\ y' = g(x, y) \end{cases}$, а $x \in (x_{min}, x_{max})$, $y \in (y_{min}, y_{max})$.

Користићемо динамички систем из претходног примера (4.2) како бисмо приказали употребу наредбе *StreamPlot*:

```
In[3] := StreamPlot[{-x + 3y, 5x - 3y}, {x, -5, 5}, {y, -5, 5}]
Out[3] :=
```



Слика 4.14: Фазни портрет динамичког система (4.2) добијен коришћењем наредбе *StreamPlot*

Навешћемо још неколико наредби које ће нам послужити у даљем раду:

- Наредба *Solve* - се користи за аналитичко решавање једначине и система једначина;
- Наредба *NDSolve* - се користи за нумеричко решавање диференцијалне једначине и система диференцијалних једначина (употребљава се када се аналитичко решење диференцијалне једначине не може лако добити);
- Наредба *ParametricPlot* - црта криве где су x и y координате дате функцијама $x(t)$ и $y(t)$, које зависе од параметра t ;
- Наредба *Table* - користи се за прављење табеле односно листе вредности задатих израза у зависности од променљивих;
- Наредба *Show* - служи за приказивање више графичких елемената (слика или графика) на једној слици.

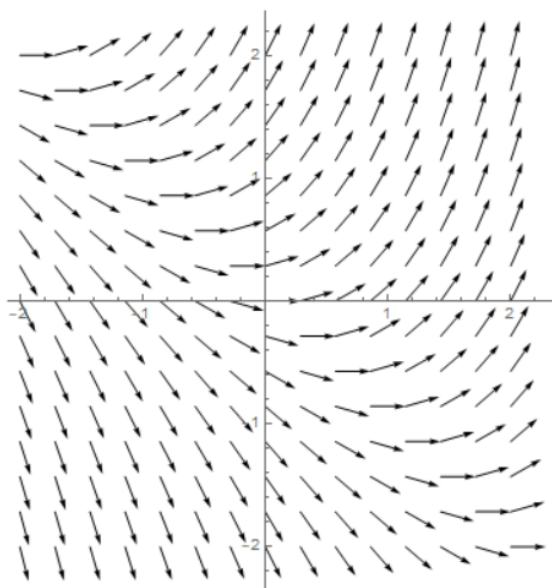
Глава 5

Решавање диференцијалних једначина првог реда користећи програмски пакет *Wolfram Mathematica*

ПРИМЕР 12. Урадити пример 1. помоћу програма *Wolfram Mathematica*.

Поље праваца за диференцијалне једначине даје геометријско понашање решења диференцијалне једначине. Користећемо *VectorFieldPlot* наредбу.

```
In[1] := p1 = VectorFieldPlot[{1, x + y}, {x, -2, 2}, {y, -2, 2}, Axes → True,  
ScaleFunction → (1&)]  
Out[1] =
```



Слика 5.1: Поље праваца диференцијалне једначине $y' = x + y$ добијен коришћењем наредбе *VectorFieldPlot*

Добијени резултат смо означили са $p1$. Опција $Axes \rightarrow True$ користи се да се x , y оса прикаже на нашем графику. Опцијом $ScaleFunction \rightarrow (1\&)$ подесили смо да величина сваког дела тангентне линије буде 1.

Више о овим опцијама и остале опције могу се наћи на $Options[VectorFieldPlot]$.

Опште решење ове диференцијалне једначине добија се коришћењем наредбе $DSolve$:

```
In[2] := sol = DSolve[y'[x] == x + y[x], y[x], x]
Out[2] = {{y[x] -> -1 - x + e^x C[1]}}
```

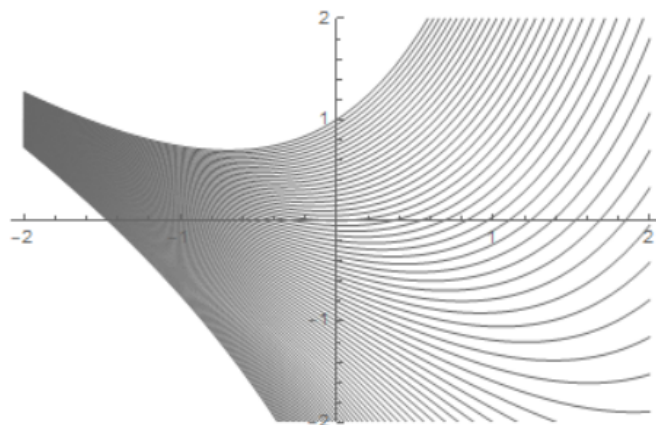
С обзиром на то да ће нам у даљем раду бити потребно опште решење и овде смо означили добијени резултат са sol . Битно је нагласати да нам $Out[2]$ враћа sol у облику сложене структуре која садржи листу решења и да ако позовемо $sol[[1, 1, 1]]$ приказаће нам $y[x]$, а аналитички израз решења $y[x] = -1 - x + e^x \cdot C[1]$ добијамо позивом $sol[[1, 1, 2]]$. Симбол $C[1]$ представља константу C у решењу $y = -1 - x + e^x \cdot C$. Како бисмо нацртали графике решења за произвољне константе, користимо наредбу $Table$. Опцијом $ReplaceAll$ односно $(/.)$ замењује се $C[1]$ са датим вредностима m у добијеном решењу sol , где нам m узима вредности између -2 и 2 са кораком 0.05 . Резултат је табела функција које ћемо означити са tp .

Напомена 1. Уколико је на крају наредбе стављен знак ; та наредба ће се успешно извршити, али се неће приказати у $Output$ -у.

```
In[3] := tp = Table[(sol[[1, 1, 2]])/.C[1] -> m, {m, -2, 2, 0.05}];
```

Користићемо $Plot$ да представимо графике ових функција и означимо са $p2$.

```
In[4] := p2 = Plot[tp, {x, -2, 2}, PlotRange -> {-2, 2},
PlotStyle -> {{GrayLevel[0.4], Thickness[0.0001]}}]
Out[4] =
```



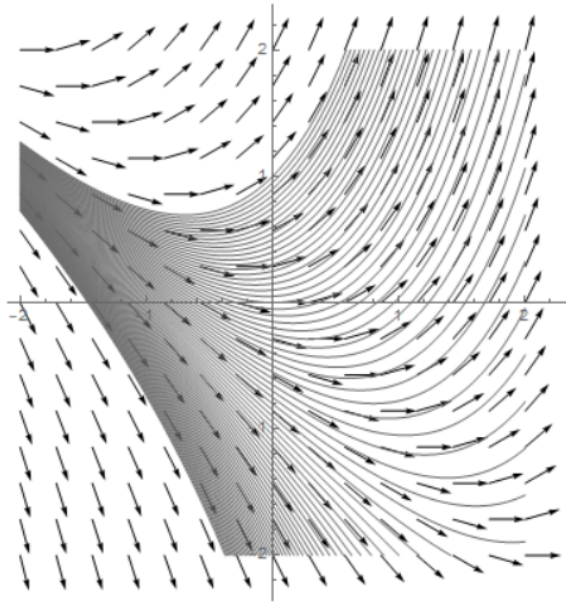
Слика 5.2: Интегралне криве диференцијалне једначине $y' = x + y$

Опција $PlotRange \rightarrow \{-2, 2\}$ подешава да y може бити у сегменту $[-2, 2]$, док инструкцијом $PlotStyle \rightarrow \{\{GrayLevel[0.4], Thickness[0.0001]\}\}$ одређујемо да графици

функција буду приказани у тамно сивој боји, ширине 0.0001.

Наредбом *Show* на истом графику приказаћемо и поља праваца и интегралне криве:

```
In[5] := Show[p1, p2]
Out[5] =
```



Слика 5.3: Интегралне криве диференцијалне једначине $y' = x + y$ добијене коришћењем наредбе *VectorFieldPlot*

Приметимо да се може предвидети понашање решења ове диференцијалне једначине посматрањем поља праваца.

ПРИМЕР 13. Решити Кошијев проблем (5.1) аналитички, а онда и коришћењем наредбе *DSolve*.

Нека је дат Кошијев проблем:

$$y'(x) = \frac{y(x)}{x}, \quad y(-3) = \frac{1}{3}. \quad (5.1)$$

Једно тривијално решење ове диференцијалне једначине је $y(x) = 0$, за $x \neq 0$. С обзиром на то да ово решење не задовољава почетни услов, посматрајмо сада да $y \neq 0$ како бисмо нашли остала решења диференцијалне једначине:

$$\frac{y'(x)}{y(x)} = \frac{1}{x}.$$

Када интегралимо обе стране, добија се:

$$\int \frac{y'(x)}{y(x)} dx = \int \frac{1}{x} dx.$$

Увођењем смене $y(x) = t$, $y'(x)dx = dt$ добијамо:

$$\int \frac{dt}{t} = \int \frac{1}{x} dx,$$

тј.

$$\ln |y(x)| = \ln |x| + c,$$

односно,

$$|y(x)| = |x| \cdot c_1, c_1 > 0.$$

Фамилија решења наше диференцијалне једначине је:

$$y(x) = x \cdot c_2, c_2 \in \mathbb{R} \setminus \{0\}.$$

Ако придружимо и тривијално решење, добија се фамилија решења која представља опште решење диференцијалне једначине $y'(x) = \frac{y(x)}{x}$:

$$y(x) = x \cdot c_3, c_3 \in \mathbb{R}, x \neq 0.$$

Заменом координата тачке $(-3, \frac{1}{3})$, уместо x и y добија се:

$$\frac{1}{3} = (-3) \cdot c_3,$$

па следи да је $c_3 = -\frac{1}{9}$. Заменом ове вредности уместо c_3 у опште решење, добија се решење Кошијевог проблема:

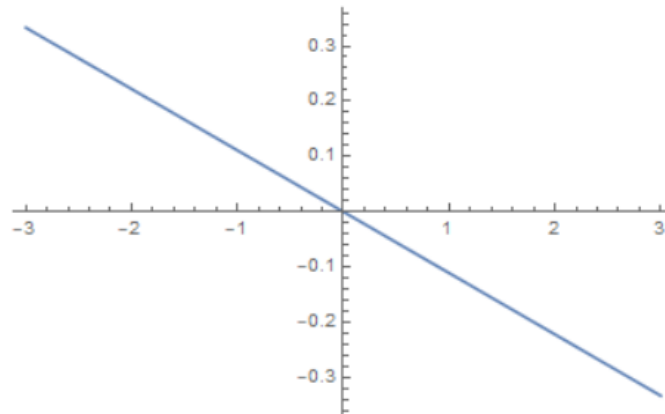
$$y(x) = -\frac{1}{9}x, x \in (-\infty, 0).$$

Решење претходно поменутог Кошијевог проблема добијено коришћењем наредбе *DSolve* је:

```
In[1] : sol = DSolve[{y'[x] == y[x]/x, y[-3] == 1/3}, y[x], x]
Out[1] = {{y[x] -> -x/9}}
```

Употребом наредбе *Plot*, нацртаћемо график решења Кошијевог проблема (5.1):

```
In[2] := Plot[sol[[1, 1, 2]], {x, -3, 3}]
Out[2] =
```



Слика 5.4: График решења Кошијевог проблема (5.1)

Напомена 2. График решења који генерише *Wolfram Mathematica* игнорише услов да је $x \neq 0$ у диференцијалној једначини, осим ако експлицитно не наведемо другачије користећи опцију $\{Exclusions \rightarrow \{x == 0\}\}$. Због тога нам се решење добијено аналитички разликује. Како мора да важи $x \neq 0$ и да решење Кошијевог проблема задовољава почетни услов, онда мора бити $x \in (-\infty, 0)$.

5.1 Примена диференцијалних једначина

Диференцијалне једначине представљају једну од најзначајнијих грана математике када се посматрају кроз призму примена. Законе физике можемо формулисати помоћу диференцијалних једначина. Такође, диференцијалне једначине имају широку примену и у медицини, биологији, хемији, економији и другим областима.

Одређену појаву у природи или физичко стање можемо описати путем диференцијалних једначина. Такав поступак назива се моделирање.

Сада ћемо истаћи неколико примера који илуструју разнолике могућности примене диференцијалних једначина.

5.2 Њутнов закон хлађења

Диференцијалне једначине првог реда могу послужити за решавање различитих проблема који су везани за температуру. На пример, медицински истражитељ може утврдити време смрти у случају убиства, хемичар може одредити време потребно да се одређена смеша охлади, а инжењер може одредити температуру при дизајнирању система за хлађење или грејање одређене просторије. Иако су различити, сваки од ових проблема зависи од основног принципа, Њутновог закона хлађења.

Њутнов закон хлађења гласи: Брзина промене температуре тела T које се хлади пропорционална је разлици температуре тела и константне температуре околине T_s . Њутнов закон хлађења моделира се помоћу диференцијалне једначине првог реда са почетним условом:

$$\frac{dT}{dt} = -k(T - T_s), \quad T(0) = T_0, \quad (5.2)$$

где је $k > 0$ коефицијент пропорционалности, а T_0 почетна температура тела (тј. температура тела у тренутку $t = 0$).

Посматрајмо сада једначину:

$$\frac{dT}{dt} = -k(T - T_s),$$

односно:

$$\frac{T'(t)}{T(t) - T_s} = -k,$$

где је $T'(t) = \frac{dT}{dt}$.

Када интегралимо обе стране, добија се:

$$\int \frac{T'(t)}{T(t) - T_s} dt = \int -k dt,$$

тј.

$$\ln |T(t) - T_s| = -kt + c, \quad c \in \mathbb{R}.$$

Односно

$$|T(t) - T_s| = c_1 \cdot e^{-kt}, \quad c_1 > 0,$$

па је:

$$T(t) = c_1 \cdot e^{-kt} + T_s, \quad c \in \mathbb{R} \setminus \{0\}. \quad (5.3)$$

Ако у претходну једначину уврстимо да је $T(0) = T_0$ онда се добија:

$$T_0 = c_1 + T_s,$$

одакле следи да је:

$$c_1 = T_0 - T_s.$$

Заменом c_1 у једначину (5.3) добијамо:

$$T(t) = (T_0 - T_s) \cdot e^{-kt} + T_s, \quad (5.4)$$

што је решење Кошијевог проблема (5.2).

Ако у програмском пакету *Wolfram Mathematica* позовемо наредбу *DSolve* за Њу-тнов закон хлађења (5.2), добићемо следеће:

```
In[1] := DSolve[{T'[t] == -k(T[t] - Ts), T[0] == Tnula}, T[t], t]
Out[1] = {{T[t] -> e^{-kt}(Tnula - Ts + e^{kt}Ts)}}
```

Може се приметити да како време пролази, тј. када t тежи бесконачности, онда је

$$\lim_{t \rightarrow +\infty} e^{-kt} = 0,$$

тада је

$$\lim_{t \rightarrow +\infty} T(t) = T_s$$

односно температура тела тежи температури околине.

ПРИМЕР 14. Пита је извађена из рерне температуре 175°C и постављена је да се хлади у просторији са температуром 20°C . После 7 минута, температура пите била је 65°C . Одредити време потребно да се пита охлади на температуру од 25°C како би могла бити поједена?

Унесимо почетне вредности $T[0] = 175$, $T_s = 20$:

```
In[1] := dj = DSolve[{T'[t] == -k(T[t] - 20), T[0] == 175}, T[t], t]
Out[1] = {{T[t] -> 5e^{-kt}(31 + 4e^{kt})}}
```

Дакле,

$$T(t) = 5 \cdot e^{-kt}(31 + 4e^{kt}) \quad (5.5)$$

С обзиром на то да је после 7 минута температура била $T(7) = 65^\circ\text{C}$, из добијене једначине $65 = 5 \cdot e^{-7k}(31 + 4e^{7k})$, следи да је $k = 0.1767$.

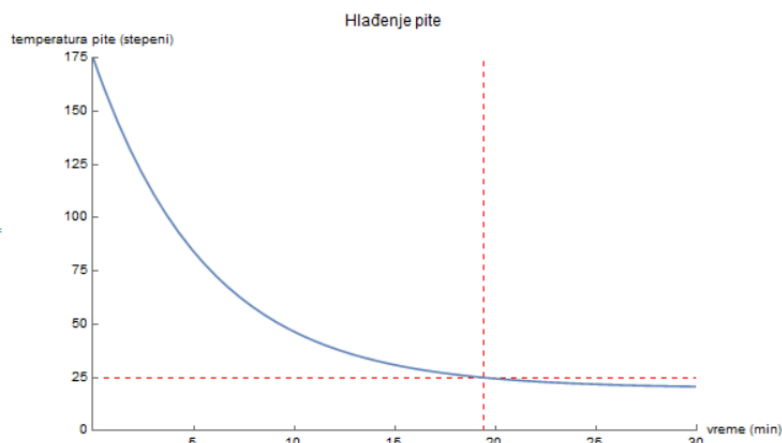
Сада одредимо колико времена је потребно да се пита охлади на температуру од 25°C . Можемо искористити наредбу *Solve* помоћу које ћемо одредити вредност променљиве t за коју је $T(t) = 25$. *Wolfram Mathematica* чува све претходне унесене и добијене вредности, тако да ће поновним позивом *dj*, t бити израчунато из једначине *Out[1]*, где постављамо да је $T[t] = 25$:

```
In[2] := timeToCooling = Solve[T[t] == 25/.dj, t]
Out[2] = {{t -> 19.434}}
```

Потребно време да се пита охлади на тражену температуру је приближно 20 минута. На слици 5.5 приказан је график функције $T(t)$ у односу на време t .

Опцијом *AxesLabel* и *PlotLabel*, додељује се назив координатама и целом графику. *GridLines* опцијом истакли смо вредност на графику када ће пита достићи тражену температуру, дакле у пресеку две праве ($x = 19.434$ и $y = 25$) које су стилизоване црвеним испрекиданим линијама.

```
In[3] := Plot[T[t]/.dj, {t, 0, 30},
  AxesLabel -> {"vreme(min)", "temperatura pite(stepeni)"}, PlotLabel -> "Hlađenje pite",
  GridLines -> {{timeToCooling}, {25}}, GridLinesStyle -> Directive[Red, Dashed]
Out[3] =
```



Слика 5.5: График функције (5.5), промена температуре пите кроз време

5.3 Закон радиоактивног распада

Вилард Либи¹ проучавао је радиоактивне елементе и открио начин за одређивање старости органских материјала помоћу радиоактивног изотопа угљеника C^{14} . Када животиње или биљке угину, оне престају да уносе свеж угљеник C^{14} . Време полураспада C^{14} је 5730 ± 40 година. Године 1960. Либи је добио Нобелову награду за своју методу утврђивања старости. Датирање радиоактивним угљеником испоставило се поуздано и тачно када су упоредили са тестовима урађеним над секвојама и датумима добијених из њихових прстена.

Закон радиоактивног распада гласи:

$$\frac{dN(t)}{dt} = -\lambda \cdot N(t), \quad (5.6)$$

где је са $N(t)$ означен број нераспаднутих атома у тренутку t , а $\lambda > 0$ константа радиоактивног распада (тј. коефицијент распадања). То значи да је брзина којом ће се атоми распадути пропорционална количини нераспаднутих атома у тренутку t . Једначину $N'(t) = -\lambda \cdot N(t)$ записаћемо на следећи начин:

$$\frac{N'(t)}{N(t)} = -\lambda.$$

Када интегралимо обе стране, добија се:

$$\ln N(t) = -\lambda \cdot t + c, \quad c \in \mathbb{R}$$

тј.

$$N(t) = e^{-\lambda \cdot t} \cdot c_1, \quad c_1 > 0.$$

¹Вилард Френк Либи (1908 – 1980), амерички физички хемичар

У почетном тренутку је $N_0 = N(0) = c_1$, па се претходно решење може записати:

$$N(t) = N_0 \cdot e^{-\lambda t}. \quad (5.7)$$

Радиоактивне супстанце се често карактеришу преко времена полураспада $T_{1/2}$. Када у (5.7) заменимо $N(T_{1/2}) = \frac{N_0}{2}$, добија се

$$\lambda = \frac{\ln 2}{T_{1/2}}.$$

Одатле следи да је:

$$N(t) = N_0 \cdot 2^{-\frac{t}{T_{1/2}}}. \quad (5.8)$$

ПРИМЕР 15. У археолошком налазишту у Египту утврђено је да дрво од кога је направљен саркофаг садржи 63% атома угљеника C^{14} . Колика је старост дрвета од кога је направљен саркофаг?

У почетном тренутку саркофаг садржи 100% угљеника, тј. $N(0) = 100\% = 1$. Претходно је речено да је време полураспада угљеника 5730 ± 40 . Заменом ових вредности у једначину (5.8) добија се:

$$N(t) = 1 \cdot 2^{-\frac{t}{5730}}.$$

Како је у тренутку t , $N(t) = 63\%$, следи:

$$0.63 = 1 \cdot 2^{-\frac{t}{5730}}.$$

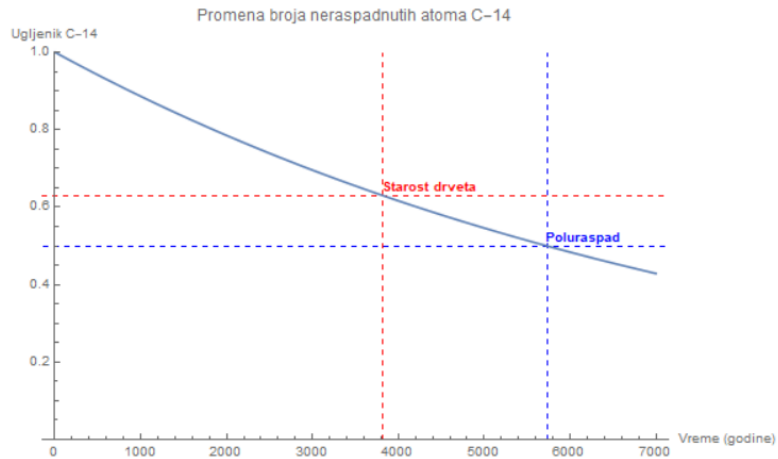
Решавањем ове једначине добија се $t \approx 3819.48$.

У програмском пакету *Wolfram Mathematica*, можемо искорисити наредбу *Solve* за решавање обичне једначине. Претходно ћемо унети дате вредности:

```
In[1] := N[0] = 1;
In[2] := N[t] = 0.63;
In[3] := z = N[t] == N[0] * Exp[-Log[2]/5730 * t];
In[4] := vreme = Solve[z, t]
Out[4] = {{t -> 3819.48}}
```

Користећи дату наредбу, одмах долазимо до тражег времена t . Промена броја нераспаднутих атома угљеника C^{14} представљена је на слици 5.6. У пресеку правих, плавом бојом истакли смо тренутак полураспада угљеника, док је црвеном бојом означена старост дрвета са 63% преосталог угљеника.

```
In[5] := Plot[z, {t, 0, 7000}, AxesLabel -> {"Vreme(godine)", "Ugljenik C - 14"},  
PlotLabel -> "Promena broja neraspadnutih atoma C - 14",  
GridLines -> {{5730, 0.5, Blue}, {vreme, 0.63, Red}}, GridLinesStyle -> Directive[Dashed],  
Text[Style["Starost drveta", Red, Bold], {vreme, 0.63}], Text[Style["Poluraspad", Blue,  
Bold], {5730, 0.5}]  
Out[5] =
```



Слика 5.6: График функције (5.8), промена броја нераспаднутих атома

Глава 6

Примена система диференцијалних једначина

6.1 Динамички модел Предатор - плен

Предатор - плен модел је пример динамичког система који се често користи у биологији или екологији за проучавање односа између две повезане популације, где једна (плен) служи као храна другој (предатор). Овај модел описује како се број јединки у обе популације мења током времена, узимајући у обзир факторе као што су стопа рађања, стопа смртности и интеракције између предатора и плена.

Један од најпознатијих основних модела овог типа је *Lotka – Volterra*¹ модел, који је представљен динамичким системом:

$$\begin{aligned}\frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= -\gamma y + \delta xy,\end{aligned}\tag{6.1}$$

где је са $x = x(t)$ обележен број јединки плена, а са $y = y(t)$ број јединки предатора у тренутку t , α , β , γ и δ су позитивне константе које представљају различите факторе. Константа α је константа рађања популације плена, односно у одсуству предатора важило би:

$$\frac{dx}{dt} = \alpha x,$$

а γ је константа смртности предатора што значи да је

$$\frac{dy}{dt} = -\gamma y$$

у одсуству плена.

Осим ових фактора, број интеракција између предатора и плена утиче на број чланова у обе популације. У првој једначини динамичког система (6.1) константа $-\beta$

¹Lotka-Volterra модел носи назив по математичарима - Alfred Lotka (1880 — 1949), амерички математичар, хемичар и статистичар и Vito Volterra (1860 — 1940), италијански математичар и физичар

представља негативну интеракцију између плена и предатора, јер контакт са преда-
тором смањује број плена, док у другој једначини са δ је означена константа која
представља позитивну интеракцију, тј. утиче на повећање броја предатора.
Важно је напоменути да *Lotka – Volterra* модел има своја ограничења, и да стварни
екосистеми могу бити далеко комплекснији, али овај модел пружа основни увид у
динамику односа предатора и плена.

ПРИМЕР 16. *Популација лосова 1990. бројала је 780 пре увођења сивих вукова, док је 5 година касније њихов број износио 1800.*

а) *Колико износи стопа рађања, пре увођена сивих вукова?*

б) *Године 1995. враћен је 21 вук и њихов број је порастао. Ако следећи динамички систем представља динамички модел лосова и вукова, графички представити број јединки ове две популације током времена и наћи фазне трајекторије посматраног динамичког система:*

$$\begin{aligned} \frac{dx}{dt} &= 0.167x - 0.08xy \\ \frac{dy}{dt} &= -0.6y + 0.05xy \\ x(0) &= 1.8, \quad y(0) = 0.021 \end{aligned} \tag{6.2}$$

а) С обзиром на то да немамо предаторе односно сиве вукове, за лосове важи:

$$\frac{dx}{dt} = \alpha x.$$

Дати услови су: $x(0) = 0.78$, и $x(5) = 1.8$, где је x број лосова дато у хиљадама у тренутку t . Искористићемо наредбу *DSolve* за решавање овог Кошијевог проблема:

$$\begin{aligned} \frac{dx}{dt} &= \alpha x \\ x(0) &= 0.78. \end{aligned} \tag{6.3}$$

```
In[1] := eqn = {x'[t] == α * x[t], x[0] == 0.78};
In[2] := DSolve[eqn, x, t]
Out[2] = {{x -> Function[{t}, 0.78e^{αt}]}};
```

Како је $x(5) = 1.8$, добија се да је стопа рађања:

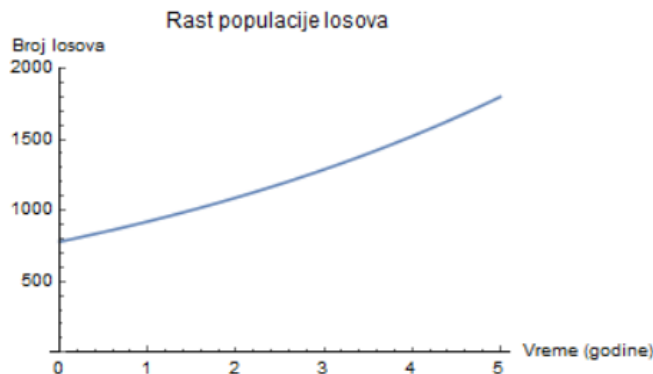
$$\alpha = 0.167.$$

Коришћењем наредбе *Solve* долазимо до истог решења:

```
In[3] := Solve[x[5] == 0.78 * Exp[α * 5], α]
Out[3] = {{α -> 1/5 Log[1.8/0.78]}}
```

Нацртајмо уз помоћ наредбе *Plot* експоненцијалну функцију раста популације лосова.

```
In[4] = los = x[t] == 0.78 * Exp[α * t];
In[5] := Plot[los, {t, 0, 5}, AxesLabel → {"Vreme (godine)", "Broj losova"},
PlotLabel → "Rast populacije losova"]
Out[5] =
```



Слика 6.1: График функције $x(t)$ Кошијевог проблема (6.3)

b) У програмском пакету *Wolfram Mathematica* дефинишимо дати динамички систем и означимо га са *eqns*. За почетни тренутак узимамо време кад су враћени вукови, то јест када је $t = 0$ имамо $x[0] = 1.8$ лосова дато у хиљадама и $y[0] = 0.021$ вукова.

```
In[4] = eqns = {x'[t] == 0.167 * x[t] - 0.08 * x[t] * y[t], y'[t] == -0.6 * y[t] + 0.05 * x[t] * y[t], x[0] == 1.8, y[0] == 0.021};
```

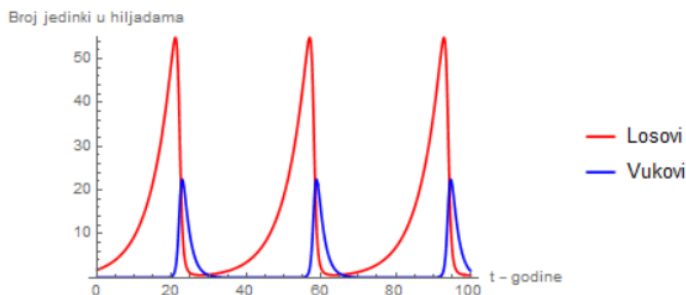
Овакав динамички систем решићемо наредбом *NDSolve*.

```
In[5] := resenje = NDSolve[eqns, {x, y}, {t, 0, 100}];
```

Како бисмо приказали график динамичког система (6.2) употребићемо наредбу *Plot*. Црвеном бојом означени су лосови у хиљадама у јединици времена t (t је дато у годинама), док су плавом бојом означени вукови.

Може се уочити на датом графику да популација лосова не изумире увођењем сивих вукова, али да сиви вукови свакако утичу на број лосова, као и да опстанак сивих вукова зависи од броја лосова.

```
In[6] := Plot[{x[t], y[t]}/.resenje, {t, 0, 100}, PlotLegends → {"Losovi", "Vukovi"},
AxesLabel → {"t - godine", "Broj jedinki u hiljadama"}, PlotStyle → {Red, Blue}]
Out[6] =
```



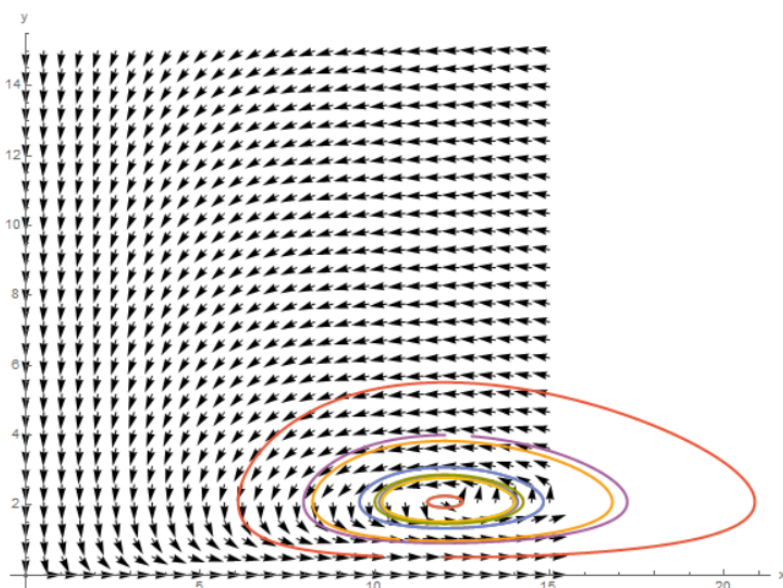
Слика 6.2: График функција $x(t)$, $y(t)$ динамичког система (6.2)

Користећемо наредбу *VectorFieldPlot* како бисмо приказали поље праваца овог динамичког система:

```
In[7] := s1 = VectorFieldPlot[{0.167x - 0.08xy, -0.6y + 0.05xy}, {x, 0, 15}, {y, 0, 15},
  Axes -> Automatic];
```

Већ смо рекли да овакав динамички систем нумерички решавамо користећи наредбу *NDSolve*. У зависности од изабраних почетних услова, наредба *Table* генерише листу добијених вредности за свако i из скупа тачака - *pocetniUslov*. На истом графику уз помоћ наредбе *Show* цртамо поље праваца и неколико фазних трајекторија.

```
In[8] := pocetniUslov = {{x[0] == 12.5, y[0] == 2.08}, {x[0] == 11.5, y[0] == 2.08},
  {x[0] == 12, y[0] == 1.5}, {x[0] == 13, y[0] == 3}, {x[0] == 14, y[0] == 2.08},
  {x[0] == 12, y[0] == 0.5}, {x[0] == 10, y[0] == 2.08}};
In[9] := resenja = Table[NDSolve[{eqns, i}, {x, y}, {t, 0, 50}], {i, pocetniUslov}];
In[10] := Show[s1, ParametricPlot[Table[x[t], y[t]/.sol, {sol, resenja}], {t, 0, 15}]]
Out[10] =
```

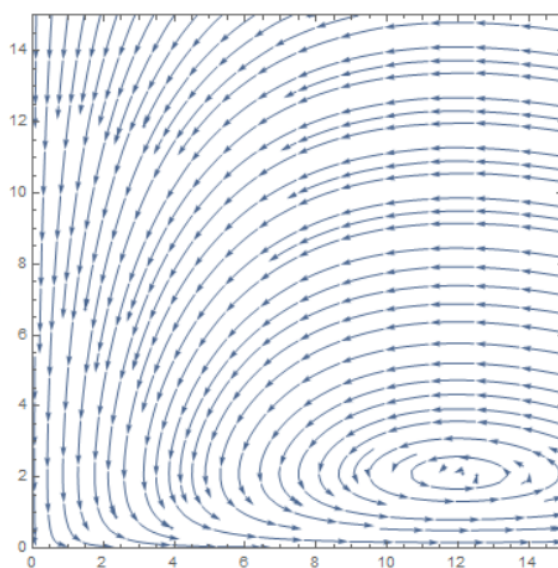


Слика 6.3: Поље праваца динамичког система (6.2)

Уочава се да сва решења осцилују око еквилибријума $E(12, 2.08)$. Ова решења откривају однос између популације лосова и популације вукова. Ако бисмо пратили једну фазну трајекторију у смеру супротном од казаљке на сату, приметили би да како расте број лосова, повећава се број сивих вукова све док популација вукова не постане пренасељена. У том тренутку популација лосова је премала за опстанак сивих вукова и тада број вукова креће да опада. Онда та популација вукова постаје мала да би контролисала плен, што опет доводи до пораста броја лосова и тај циклус се понавља.

Фазни портрет добијамо наредбом *StreamPlot*:

```
In[12] := StreamPlot[{0.167x - 0.08xy, -0.6y + 0.05xy}, {x, 0, 15}, {y, 0, 15}]
Out[12] =
```



Слика 6.4: Фазни портрет динамичког система (6.2)

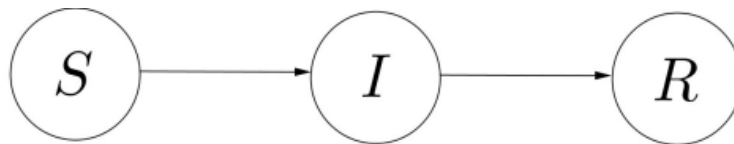
6.2 Динамички модел инфективних болести SIR

Посматра се популација са фиксираним бројем људи n . Таква популација подељена је у три одвојене групе: осетљиви, заражени, имуни. Прву групу карактеришу они људи који немају развијени имунитет, односно они који се још увек нису заразили. За ову групу користи се ознака S , од енглеске речи *susceptible* што у преводу значи осетљив. Другу групу чине инфицирани људи, енг. *infected* и користи се ознака I , док у трећу групу спадају они људи који су прележали болест тј. стекли су имунитет (енг. *recovered*). Ознака ове групе је R . Па се овај динамички модел назива *SIR*.

Овај модел представља прелазак из једне у другу групу. Осетљива особа када се зарази, прелази у групу заражених, а касније у групу опорављених односно имуних. Тај прелазак је приказан на следећој слици:

Број чланова сваке групе је променљива зависна од времена t .

$S(t)$ - број осетљивих људи у тренутку t .



Слика 6.5: SIR модел

$I(t)$ - број заражених људи у тренутку t .

$R(t)$ - број људи који су прележали болест у тренутку t .

Модел је представљен следећим системом диференцијалних једначина:

$$\begin{aligned} S'(t) &= -\beta \cdot S(t) \cdot I(t) \\ I'(t) &= \beta \cdot S(t) \cdot I(t) - \alpha \cdot I(t) \\ R'(t) &= \alpha \cdot I(t), \end{aligned} \tag{6.4}$$

где је са β означена стопа заражавања тј. преносивости болести, а са α стопа уклањања заразе и $\alpha, \beta \geq 0$.

Претпоставке под којима важи овај модел су:

- Болест је краткотрајна, односно морталитет и наталитет немају утицаја на популацију, па је n фиксиран број људи;
- Не постоји могућност да се особа која је прележала болест поново зарази;
- Популација је затвореног типа током трајања епидемије, тј. не узимамо у обзир миграцију становништва;
- Једни начин да осетљив напусти групу јесте да се зарази;
- Под појмом имуни сматрамо оне који су прележали болест.

Како важи и претпоставка да је сваки члан бар у једној од три групе онда у сваком тренутку важи:

$$S(t) + I(t) + R(t) = n.$$

Због тога последња једначина из модела (6.4) се може изоставити, јер уколико је познат број чланова у групи $S(t)$ и у групи $I(t)$, лако се може закључити колико има чланова у групи $R(t)$.

Просечан члан остварује βn контаката, тако да сваки инфицирани долази у контакт са βn чланова популације. Вероватноћа да дође у контакт баш са осетљивом особом је $\frac{S}{n}$. Због тога $\beta n \frac{S}{n} I = \beta SI$ је први сабирак у другој једначини. Одатле следи да када се број чланова групе осетљивих смањује, тада се повећава број чланова групе заражених.

Уколико би, у почетном тренутку $t = 0$, важило $I(0) = 0$, не би било болести. Дакле, у почетном тренутку мора да важи:

$$I(0) = I_0, 0 < I_0 < n$$

$$S(0) = S_0$$

$$R(0) = R_0 = 0.$$

Број људи који су прележали болест јесте нула у почетку самог ширења инфективне болести.

Ако посматрамо сада другу једначину из модела (6.4), у тренутку $t = 0$ важиће:

$$\frac{dI}{dt}\Big|_{t=0} = \beta S_0 I_0 - \alpha I_0.$$

Можемо разматрати два случаја: $\beta S_0 I_0 - \alpha I_0 < 0$ или $\beta S_0 I_0 - \alpha I_0 > 0$. Како у почетном тренутку важи да је $I_0 \neq 0$, обе ове неједначине смемо да поделимо са I_0 , тако да је $\beta S_0 - \alpha < 0$ или $\beta S_0 - \alpha > 0$.

Тј. $S_0 < \frac{\alpha}{\beta}$ или $S_0 > \frac{\alpha}{\beta}$. Односно:

$$\frac{\beta}{\alpha} \cdot S_0 < 1 \quad \text{или} \quad \frac{\beta}{\alpha} \cdot S_0 > 1.$$

Означимо са:

$$M_0 = \frac{\beta}{\alpha} \cdot S_0,$$

M_0 је број који представља колико просечно особа популације може да зарази инфицирана особа. Уколико је $M_0 > 1$ онда долази до епидемије, тј. врло брзо долази до појаве великог броја инфицираних. Ако је $M_0 < 1$, онда не долази до појаве епидемије, односно болест нестаје из популације. И трећа могућност ако је $M_0 = 1$ у питању је ендемија, тј. болест је стално присутна у популацији са малим бројем инфицираних који временом остаје приближно непроменљив.

ПРИМЕР 17. *Заразна болест се шири у малој заједници, са фиксираним бројем људи $n = 10$, контактом заражених појединаца. Претпоставља се да ниједан појединац не напусти заједницу. Нека је стопа заражавања једнака $\beta = 0.2$, а стопа опоравка $\alpha = 0.7$. У ком случају ће доћи до наглог пораста броја оболелих? Графички представити овај SIR модел.*

Претходно смо показали да уколико је $M_0 > 1$ имаћемо епидемију, односно у нашем случају, брзо ширење заразне болести:

$$\frac{\beta}{\alpha} \cdot S_0 > 1.$$

Када дате параметре из задатка уврстимо у претходну неједначину добија се:

$$\frac{2}{7} \cdot S_0 > 1,$$

односно

$$S_0 > \frac{7}{2}.$$

Дакле, закључујемо да уколико је број осетљивих људи на почетку заразе већи од 3 имаћемо неку врсту локализоване „епидемије“.

Унећемо даге вредности у наш програм:

```
In[1] :=  $\beta = 0.2;$ 
In[2] :=  $\alpha = 0.7;$ 
In[3] := eqnS = S'[t] ==  $-\beta * S[t] * I[t];$ 
In[4] := eqnI = I'[t] ==  $\beta * S[t] * I[t] - \alpha * I[t];$ 
In[5] := eqnR = R'[t] ==  $\alpha * I[t];$ 
```

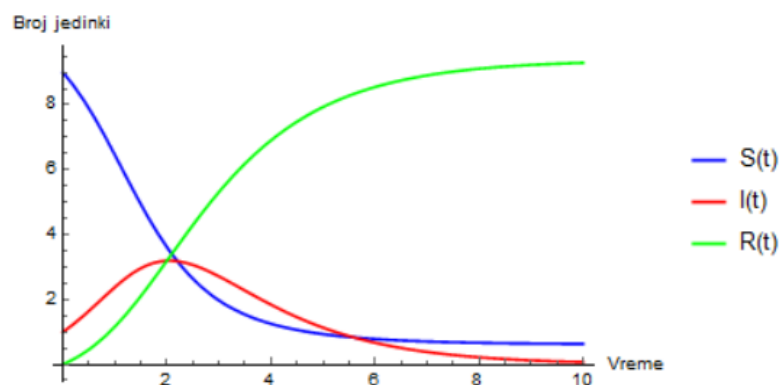
У почетном тренутку важи да $R[0] = 0$, претпоставимо да у том почетном тренутку имамо једну заражену особу. Тј. $I[0] = 1$, одатле ће следити да је: $S[0] = n - 1 = 9$. И у овом примеру користићемо наредбу *NDSolve*:

```
In[6] := sol = NDSolve[{eqnS, eqnI, eqnR, S[0] == 9, I[0] == 1, R[0] == 0}, {S, I, R}, {t, 0, 10}]
Out[6] =
```

```
{ { S → InterpolatingFunction [ Domain: {{0., 10.}}, Output: scalar ],
  I → InterpolatingFunction [ Domain: {{0., 10.}}, Output: scalar ],
  R → InterpolatingFunction [ Domain: {{0., 10.}}, Output: scalar ] }
```

Искористимо сада наредбу *Plot* за приказ графика функција $S(t)$, $I(t)$, $R(t)$:

```
In[7] := Plot[{S[t], I[t], R[t]}/.sol, {t, 0, 10},
PlotLegends → {" S(t)", " I(t)", " R(t)"}, AxesLabel → {" Vreme", " Broj jedinki"},
PlotStyle → {Blue, Red, Green}, AxesStyle → Black]
Out[7] =
```



Слика 6.6: Графици функција $S(t)$, $I(t)$, $R(t)$

Број осетљивих људи је опадајућа функција, док број инфицираних видимо да расте, односно зараза се шири. То ће се дешавати до тренутка када број осетљивих људи падне испод 3,5 (*SIR* моделом се описује динамика ширења заразних болести па се због тога број људи може апроксимирати на \mathbb{R}). У том тренутку број инфицираних почеће да опада. Са датог графикона видимо да како време одмиче у једном тренутку неће бити више инфицираних и зараза ће ишчезнути из популације. На графикону се јасно види и да број имуних све време расте.

Глава 7

Закључак

Wolfram Mathematica може послужити за јаснији увид и разумевање самих диференцијалних једначина. Уграђене наредбе овог програма олакшавају креирање графика решења диференцијалних једначина, исцртавање поља праваца, фазних трајекторија динамичких система и друго.

У овом раду користили смо верзију 11.3 програмског пакета *Wolfram Mathematica*. Поред овог програма, постоје и други програми који се могу користити при решавању или разумевању диференцијалних једначина, попут програма *GeoGebra*.

У својим првим верзијама *Wolfram Mathematica* имао је доста ограничења и недостатака. Временом овај пакет се развијао од стране *Wolfram Research*-а, али и данас има своје недостатке. Осим комплексности интерфејса и саме синтаксе овог програма, видели смо да не мора увек дати „коректно“ решење или да нам понекад решење врати у облику који није експлицитан. Свакако има и својих предности. Може бити од велике користи као алат за визуелно представљање математичких садржаја у образовању.

Литература

- [1] Marta L. Abell, James P. Braselton, Differential Equations with Mathematica, California, Academic Press, third edition 2004.
- [2] W. Adkins, M. Davidson, Ordinary Differential Equations, Springer, 2012.
- [3] Clay C. Ross, Differential Equations An Introduction with Mathematica, Springer, second edition 2004.
- [4] Радоје Шћепановић, Јулка Кнежевић-Миљановић, Љубомир Протић, Диференцијалне једначине, Универзитет у Београду, Математички факултет, Београд, 2000.
- [5] Небојша Лажетић, Диференцијалне једначине А, скрипта, Универзитет у Београду, Математички факултет, Београд, 2016.
- [6] <https://www.cfm.brown.edu/people/dobrush/am33/Mathematica/intro/start.html>
- [7] <https://12000.org/>