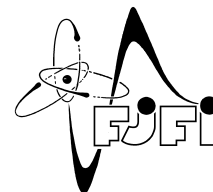


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Aplikace adjungovaných rovnic pro hledání parametrů obyčejných a parciálních diferenciálních rovnic

Application of adjoint equations for fitting of parameters of ordinary and partial differential equations

Bakalářská práce

Autor: **Jan Miesbauer**
Vedoucí práce: **doc. Ing. Tomáš Oberhuber, Ph.D.**
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Jan Miesbauer
Studijní program:	Matematické inženýrství
Studijní specializace:	Matematická informatika
Název práce (česky):	Aplikace adjungovaných rovnic pro hledání parametrů obyčejných a parciálních diferenciálních rovnic
Název práce (anglicky):	Application of adjoint equations for fitting of parameters of ordinary and partial differential equations

Pokyny pro vypracování:

- 1) Seznamte se s numerickými metodami pro řešení obyčejných diferenciálních rovnic (ODR) a reakčně difuzních parciálních diferenciálních rovnic (PDR).
- 2) Pomocí těchto metod řešte vybrané ODR jako Volterrovy-Lotkovy rovnice, SIR model, Lorentzův atraktor, apod.
- 3) Pomocí vhodných numerických metod řešte vybrané reakčně difuzní PDR jako Grayův-Scottův model, model fázového pole, apod.
- 4) Odvoďte příslušné adjungované rovnice a použijte je pro zpětný odhad parametrů vybraných modelů.
- 5) Proveďte výpočetní studii.

Doporučená literatura:

- 1) A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics. Texts in Applied Mathematics 37, Springer, 2006.
- 2) M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints. Mathematical Modelling: Theory and Application 23, Springer, 2009.
- 3) J. D. Murray, Mathematical Biology: I. An Introduction. Interdisciplinary Applied Mathematics 17, Springer, 2007.

Jméno a pracoviště vedoucího bakalářské práce:

doc. Ing. Tomáš Oberhuber, Ph.D.

Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze, Trojanova 13, 120 00 Praha 2


Jméno a pracoviště konzultanta:

Datum zadání bakalářské práce: 31.10.2022

Datum odevzdání bakalářské práce: 2.8.2023

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 31.10.2022


.....
garant oboru


.....
vedoucí katedry




.....
děkan

Poděkování:

Chtěl bych zde poděkovat především svému školiteli doc. Ing. Tomáši Oberhuberovi, Ph.D. za cenné rady a věnovaný čas při vedení mé bakalářské práce. Dále bych chtěl poděkovat své rodině a přátelům za jejich rady a trpělivost.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 2. srpna 2023

Jan Miesbauer

Název práce:

Aplikace adjungovaných rovnic pro hledání parametrů obyčejných a parciálních diferenciálních rovnic

Autor: Jan Miesbauer

Obor: Matematické inženýrství

Zaměření: Matematická informatika

Druh práce: Bakalářská práce

Vedoucí práce: doc. Ing. Tomáš Oberhuber, Ph.D. Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze, Trojanova 13, 120 00 Praha 2

Abstrakt: Tato práce se zabývá odhadem parametrů vybraných matematických modelů popsaných pomocí soustav obyčejných a parciálních diferenciálních rovnic z dat. K výpočtu gradientu, který použijeme v rámci optimalizace ztrátové funkce, představíme odvození adjungované rovnice, kterou v práci pro jednotlivé modely odvodíme. Pro srovnání dále porovnáme tuto metodu s přímým výpočtem gradientu pomocí metody konečných diferencí. Výsledné rovnice řešíme pomocí numerických metod implementovaných v C++. Nakonec provedeme výpočetní studii, ve které porovnáme obě metody a jejich funkčnost na jednotlivé modely. S výjimkou případu výpočtu gradientu pomocí adjungované rovnice pro Lorenzův model jsme ukázali, že parametry modelů je možné pomocí těchto metod odhadnout.

Klíčová slova: adjungovaná rovnice, gradientní sestup, metoda konečných diferencí, optimalizace

Title:

Application of adjoint equations for fitting of parameters of ordinary and partial differential equations

Author: Jan Miesbauer

Abstract: This work deals with the estimation of parameters of selected mathematical models described by systems of ordinary and partial differential equations from data. To calculate the gradient we use in the optimization of the loss function, we present derivation of the adjoint equation that we will derive in this paper for each model. For comparison, we further compare this method with the direct calculation of the gradient using the finite difference method. We solve the resulting equations using numerical methods implemented in C++. Finally, we perform a computational study in which we compare the two methods and their performance on individual models. Except for the case of calculating the gradient using the adjoint equation for the Lorenz model we have shown that the parameters of the models can be estimated using these methods.

Key words: adjoint equation, gradient descent, finite difference method, optimization

Obsah

Úvod	7
1 Teoretický úvod	8
1.1 Optimalizace	8
1.2 Gradientní metody	9
1.3 Normalizace gradientu	10
2 Odhad parametrů pro obyčejné diferenciální rovnice	11
2.1 Odvození adjungované rovnice pro obecný model	11
2.2 Výpočet gradientu pomocí konečných diferencí	13
2.3 Odvození adjungovaných rovnic pro konkrétní modely	13
2.3.1 Volterrův-Lotkův model	13
2.3.2 SIR model	14
2.3.3 Lorenzův model	15
2.4 Numerické řešení	16
2.4.1 Dopředná rovnice	16
2.4.2 Adjungovaná rovnice	17
2.4.3 Gradient	17
2.4.4 Ztrátová funkce	18
2.5 Implementace	18
2.6 Výpočetní studie	19
2.7 Diskuze výsledků	26
3 Odhad parametrů pro Grayův-Scottův model	27
3.1 Grayův-Scottův model	27
3.2 Odvození adjungované rovnice	28
3.3 Numerické řešení	32
3.3.1 Dopředná rovnice	33
3.3.2 Adjungovaná rovnice	34
3.3.3 Gradient	35
3.3.4 Ztrátová funkce	35
3.4 Implementace	35
3.5 Výpočetní studie	36
3.6 Diskuze výsledků	39
Závěr	40

Úvod

V této práci se zabýváme možnostmi odhadu parametrů matematických modelů popsaných pomocí soustav obyčejných diferenciálních rovnic (ODR) a parciálních diferenciálních rovnic (PDR) z dat. K tomu formulujeme příslušnou optimalizační úlohu na minimalizaci ztrátové funkce, k jejíž optimalizaci použijeme metodu založenou na gradientním sestupu. Gradient v této práci vypočteme dvojím způsobem, jednak odvozením adjungované rovnice pro daný model a dále přímým výpočtem pomocí metody konečných diferencí. Výsledné algoritmy implementujeme v programovacím jazyce C++. Zdrojové kódy použité v rámci simulací jsou k dispozici zde: *zdrojové kódy*. Nakonec provedeme výpočetní studii, kde porovnáme funkčnost obou metod.

Práce je rozdělená na tři kapitoly. V první kapitole uděláme krátký přehled nástrojů z oblasti optimalizace a strojového učení, které v této práci využijeme. Druhá kapitola obsahuje odhad parametrů pro modely popsané obyčejnými diferenciálními rovnicemi. V této kapitole nejprve odvodíme adjungovanou rovnici pro obecný model. Následně představíme jednotlivé modely - Volterrův-Lotkův model, SIR model a Lorenzův model a odvodíme adjungované rovnice v jejich konkrétních případech. Na konci následuje výpočetní studie mapující schopnost algoritmů odhadnout parametry pro jednotlivé modely. Ve třetí kapitole zopakujeme tento postup pro Grayův-Scottův model reakčně-difúzních rovnic. V této kapitole odvodíme adjungovanou rovnici pouze pro tento konkrétní model. Nakonec opět provedeme výpočetní studii.

Kapitola 1

Teoretický úvod

1.1 Optimalizace

V této sekci představíme krátký teoretický přehled ze základů optimalizace [5]. Optimalizaci můžeme obecně rozdělit na optimalizaci bez vazby a optimalizaci s vazbou. Uvažujme funkci $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Jelikož dále budeme chtít použít gradientní metody, budeme požadovat alespoň $f \in C^1(\mathbb{R}^n)$. Tato funkce se v optimalizaci často nazývá účelová funkce, v této práci však budeme používat termín ztrátová funkce, který se často používá ve strojovém učení. Minimalizační úlohou pak můžeme nazvat úlohu

$$\min_{x \in \mathbb{R}^n} f(x). \quad (1.1)$$

V další části nás bude zajímat optimalizace s vazbou, v našem případě danou pomocí soustavu obyčejných nebo parciálních diferenciálních rovnic. Úlohu na hledání minima pak můžeme přepsat do tvaru

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{za podmínky } g(x) = 0, \quad (1.2)$$

kde $g(x)$ je daná obyčejná nebo parciální diferenciální rovnice. Označíme-li množinu přípustných řešení jako $M \subset \mathbb{R}^n$, kde

$$M = \{x \in \mathbb{R}^n \mid g(x) = 0\},$$

můžeme úlohu s vazbou (1.2) přepsat do tvaru

$$\min_{x \in M} f(x). \quad (1.3)$$

Poznamenejme, že v této práci nás bude zajímat argument minima dané ztrátové funkce. Formálně tedy budeme řešit úlohu

$$\operatorname{argmin}_{x \in M} f(x). \quad (1.4)$$

Předpoklady na dané funkce a používané metody však zůstanou zcela stejné.

1.2 Gradientní metody

Vzhledem k způsobu řešení úlohy zde představíme krátký přehled gradientních metod, které budeme dále v práci využívat. Je známo, že mnoho problémů z oblasti strojového učení a příbuzných oborů lze formulovat jako optimalizační úlohu. Ztrátové funkce jsou však často velmi komplikované a většinou není možné hledat jejich extrémy analyticky. Proto se v praxi často používají metody založené na využití gradientu ztrátové funkce.

Mějme funkci $f : \mathbb{R}^n \rightarrow \mathbb{R}$, kde $f \in C^1(\mathbb{R}^n)$. Gradientní sestup je iterační metoda, která pro počáteční odhad x_0 generuje posloupnost $(x_k)_{k \geq 1} \subset \mathbb{R}^n$ postupem

$$x_{k+1} = x_k - \alpha \nabla f(x_k), \quad (1.5)$$

kde $\alpha > 0$ je velikost kroku v gradientním sestupu. Je známo, že větší hodnoty α mohou vést k rychlejší konvergenci, zároveň však mohou způsobovat nestabilitu gradientního sestupu, oscilaci kolem lokálních minim nebo divergenci.

V praxi se často používají různé varianty založené na gradientním sestupu, které se snaží zlepšit jeho stabilitu a rychlost konvergence. Pro implementaci v této práci budeme konkrétně používat algoritmus RMSprop (Root Mean Squared Propagation) [1]. Ten, kromě posloupnosti bodů $(x_k)_{k \geq 1} \subset \mathbb{R}^n$, generuje i pomocnou posloupnost $(A_k)_{k \geq 1} \subset \mathbb{R}^n$ následujícím předpisem

$$A_{k+1} = \rho A_k + (1 - \rho)(\nabla f(x_k))^2, \quad (1.6)$$

$$x_{k+1} = x_k - \frac{\alpha}{\sqrt{A_{k+1}}} \nabla f(x_k), \quad (1.7)$$

kde $\alpha > 0$ je velikost kroku v gradientním sestupu a $\rho \in (0, 1)$. Vektor A_0 můžeme inicializovat například jako $A_0 = (0, \dots, 0) \in \mathbb{R}^n$. Symbolicky pak jeden krok v tomto algoritmu můžeme zapsat jako

Algoritmus 1: Krok v RMSprop algoritmu.

Vstup: α, ρ

Vstup: $A_k, \nabla f(x_k)$

- 1: $A_{k+1} = \rho A_k + (1 - \rho)(\nabla f(x_k))^2$
 - 2: $x_{k+1} = x_k - \frac{\alpha}{\sqrt{A_{k+1}}} \nabla f(x_k)$.
-

1.3 Normalizace gradientu

Jelikož po výpočtu gradientu často dostáváme v normě velmi velké hodnoty gradientu, v oblasti strojového učení a příbuzných oborech se často používají různé metody jeho normalizace. První a nejjednodušší možnost je normalizovat gradient podělením všech jeho složek jeho normou. Mějme funkci $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Vektor $\nabla f(x)$ pro $x \in \mathbb{R}^n$ pak můžeme normalizovat jako

$$(\nabla f(x))_k = \frac{(\nabla f(x))_k}{\|\nabla f(x)\|}, \quad k \in \hat{n}, \quad (1.8)$$

kde $\|\nabla f(x)\|$ je norma gradientu funkce f . Tímto způsobem se můžeme zbavit velkých hodnot normy gradientu $\nabla f(x)$, avšak častým problémem, který tento způsob neřeší, jsou velké rozdíly v absolutní hodnotě jednotlivých složek gradientu. Ty mohou v praxi negativně ovlivnit numerickou stabilitu používaných gradientních metod.

K řešení tohoto problému můžeme složky gradientu standardizovat [9] předpisem

$$(\nabla f(x))_k = \frac{(\nabla f(x))_k - \mu}{\sigma}, \quad k \in \hat{n}, \quad (1.9)$$

kde $\mu := \frac{1}{n} \sum_{k=1}^n (\nabla f(x))_k$ a $\sigma := \sqrt{\frac{1}{n} \sum_{k=1}^n (x_k - \mu)^2}$. Tento způsob sice změní směr gradientu, avšak zaručuje vyšší numerickou stabilitu, jelikož omezuje velké rozdíly v absolutních hodnotách složek gradientu.

Poslední normalizace, kterou využijeme je logaritmická transformace [9] předpisem

$$(\nabla f(x))_k = \begin{cases} \log_{10}((\nabla f(x))_k), & \text{pokud } (\nabla f(x))_k > 0, \\ -\log_{10}((-\nabla f(x))_k), & \text{pokud } (\nabla f(x))_k < 0, \end{cases} \quad (1.10)$$

kde $k \in \hat{n}$.

Během samotné simulace budeme používat kombinaci všech těchto metod. Na začátku optimalizace použijeme standardizaci nebo logaritmickou transformaci, abychom předešli výskytu velmi velkých složek gradientu. Pokud se budeme přibližovat k minimu ztrátové funkce, změníme metodu na obyčejnou normalizaci.

Kapitola 2

Odhad parametrů pro obyčejné diferenciální rovnice

2.1 Odvození adjungované rovnice pro obecný model

V této sekci ukážeme obecné odvození adjungované rovnice pro matematický model popsany pomocí systému obyčejných diferenciálních rovnic [8]. Uvažujme časový interval $[0, T] \subset \mathbb{R}$, na kterém budeme úlohu řešit a vektor parametrů $p \in \mathbb{R}^m$. Dále uvažujme obecnou počáteční úlohu tvaru

$$\dot{u}(t, p) = f(t, p, u(t, p)), \quad \text{na } [0, T] \quad (2.1)$$

$$u(0, p) = u_0, \quad (2.2)$$

kde $u : [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ je neznámá funkce, $f : [0, T] \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ je pravá strana a $u_0 \in \mathbb{R}^n$ je počáteční podmínka. Tuto rovnici budeme dále nazývat dopřednou rovnicí. Dále uvažujme skalární funkci $h = h(t, p)$ definovanou jako

$$h(t, p) := (u(t, p) - d(t))^2, \quad (2.3)$$

kde $d(t)$ je časová řada naměřených dat pocházející ze stejného modelu. Zavedeme-li ztrátovou funkci

$$J(p) := \int_0^T h(t, p) dt = \int_0^T (u(t, p) - d(t))^2 dt, \quad (2.4)$$

můžeme definovat optimalizační úlohu

$$p^* = \operatorname{argmin}_{p \in \mathbb{R}^m} J(p). \quad (2.5)$$

Vyřešením optimalizační úlohy (2.5) nalezneme parametry p^* , které generují data $d(t)$. Jelikož dále budeme chtít použít některou z metod založenou na gradientním sestupu, budeme potřebovat vypočítat gradient funkce $J(p)$ vzhledem k parametrům p .

V následující části představíme odvození adjungované rovnice jako jeden z možných způsobů, jak gradient $\nabla J(p)$ vypočítat. K tomu zavedeme Lagrangeovy multiplikátory $\lambda \in \mathbb{R}^n$ a získáme rozšířenou ztrátovou funkci $J^*(p)$ tvaru

$$J^*(p) = \int_0^T \left(h + \lambda^T (\dot{u} - f) \right) dt. \quad (2.6)$$

Poznamenejme, že pokud $\dot{u} - f = 0$, tj. splníme dopřednou rovnici, pak $J(p) = J^*(p)$.

Nyní můžeme pokračovat s výpočtem $\nabla J^*(p)$. S použitím derivace složené funkce získáme

$$\nabla J^*(p) = \int_0^T \left(\frac{\partial h}{\partial p} + \frac{\partial h}{\partial u} \frac{\partial u}{\partial p} + \lambda^T \left(\frac{\partial \dot{u}}{\partial p} - \frac{\partial f}{\partial p} - \frac{\partial f}{\partial u} \frac{\partial u}{\partial p} \right) \right) dt.$$

Jelikož se chceme vyhnout výpočtu $\frac{\partial \dot{u}}{\partial p}$, viz. porovnání metod v sekci 2.2, přepíšeme tento člen pomocí integrace per partes

$$\int_0^T \lambda^T \frac{\partial \dot{u}}{\partial p} dt = \left[\lambda^T \frac{\partial u}{\partial p} \right]_0^T - \int_0^T \frac{\partial u}{\partial p} \frac{\partial \lambda^T}{\partial t} dt.$$

Dosažením zpět a vytknutím členu $\frac{\partial u}{\partial p}$ získáme

$$\nabla J^*(p) = \int_0^T \left(\left(\frac{\partial h}{\partial u} - \lambda^T \frac{\partial f}{\partial u} - \frac{\partial \lambda^T}{\partial t} \right) \frac{\partial u}{\partial p} + \frac{\partial h}{\partial p} - \lambda^T \frac{\partial f}{\partial p} \right) dt + \left[\lambda^T \frac{\partial u}{\partial p} \right]_0^T.$$

Abychom zamezili nutnosti výpočtu členů obsahujících $\frac{\partial u}{\partial p}$, získáme následující adjungovanou rovnici s podmínkou danou v konečném čase T , kterou řešíme zpětně v čase

$$\frac{\partial \lambda^T}{\partial t} = \frac{\partial h}{\partial u} - \lambda^T \frac{\partial f}{\partial u}, \quad \text{na } [0, T] \quad (2.7)$$

$$\lambda^T \frac{\partial u}{\partial p}(T) = 0. \quad (2.8)$$

Protože funkce $h = (u-d)^2$ bude v našem případě shodná pro všechny modely, můžeme člen $\frac{\partial h}{\partial u}$ vypočítat obecně. Jednoduchým zderivováním získáme

$$\frac{\partial h}{\partial u} = 2(u-d). \quad (2.9)$$

Dále, jelikož se chceme vyhnout výpočtu $\frac{\partial u}{\partial p}$, zredukuje se podmínka (2.8) na $\lambda^T(T) = 0$. Celkem tedy můžeme úlohu pro výpočet adjungované rovnice zapsat jako

$$\frac{\partial \lambda^T}{\partial t} = 2(u-d) - \lambda^T \frac{\partial f}{\partial u}, \quad \text{na } [0, T] \quad (2.10)$$

$$\lambda^T(T) = 0. \quad (2.11)$$

Všimněme si, že (2.10) je obyčejná lineární diferenciální rovnice pro λ^T , což může komplikovat stabilitu výpočtu, podrobněji viz. diskuze v sekci 2.6.

Výpočet gradientu $\nabla J^*(p)$ se pak zjednoduší na tvar

$$\nabla J^*(p) = \int_0^T \left(\frac{\partial h}{\partial p} - \lambda^T \frac{\partial f}{\partial p} \right) dt - \lambda^T \frac{\partial u}{\partial p}(0). \quad (2.12)$$

Jelikož navíc h , ani $u(0)$ v našem případě přímo nezávisí na p , gradient se dále zjednoduší na tvar

$$\nabla J^*(p) = \int_0^T \left(-\lambda^T \frac{\partial f}{\partial p} \right) dt. \quad (2.13)$$

2.2 Výpočet gradientu pomocí konečných diferencí

Alternativním způsobem výpočtu gradientu je jeho přímý výpočet pomocí aproximace derivací konečnými diferencemi. Symbolicky můžeme k -tou složku gradientu $\nabla J(p)$ aproximovat jako

$$(\nabla J(p))_k \approx \frac{J(p + \varepsilon e_k) - J(p)}{\varepsilon}, \quad (2.14)$$

kde $\varepsilon > 0$ a e_k je k -tý vektor ze standardní báze prostoru \mathbb{R}^m . Poznamenejme, že přestože jsme tento vzorec zařadili do této kapitoly, naprosto analogický tvar výpočtu gradientu bude využíván i během výpočetní studie v další kapitole.

Srovnáme-li teoreticky obě metody, výhoda výpočtu přes adjungovanou rovnici spočívá v tom, že náročnost výpočtu gradientu nezávisí na dimenzi \mathbb{R}^m , tedy počtu parametrů. Jelikož pro libovolný počet provedeme jeden výpočet dopředné a jeden výpočet adjungované rovnice, časová náročnost je konstantní vzhledem k počtu parametrů. To je také důvod, proč se pomocí adjungované rovnice snažíme vyhnout výpočtu členů ve tvaru $\frac{\partial u}{\partial p}$ a $\frac{\partial \dot{u}}{\partial p}$, neboť jejich výpočet by vyžadoval $m + 1$ řešení rovnice (2.1).

Naproti tomu při výpočtu přes konečné diference musíme provést $m + 1$ výpočtů dopředné rovnice, a tedy časová náročnost roste lineárně vzhledem k počtu parametrů. Na druhou stranu, výhodou oproti počítání gradientu pomocí adjungované rovnice může být jednodušší implementace, jelikož stačí pracovat pouze s dopřednou rovnicí. Bližší porovnání obou metod bude následovat v sekci 2.6.

2.3 Odvození adjungovaných rovnic pro konkrétní modely

2.3.1 Volterrův-Lotkův model

Volterrův-Lotkův model [4], často známý jako systém dravec-kořist, je klasický biologický systém popisující interakci dvou soupeřících skupin, popsanych proměnnými x a y . Proměnnou x lze interpretovat jako kořist a proměnnou y jako dravce. Model má tvar

$$\begin{aligned} \frac{dx}{dt} &= \alpha_1 x - \beta_1 xy, & \text{na } [0, T] \\ \frac{dy}{dt} &= -\alpha_2 y + \beta_2 xy, \end{aligned} \quad (2.15)$$

kde $\alpha_1, \beta_1, \alpha_2, \beta_2 > 0$ a $p = (\alpha_1, \beta_1, \alpha_2, \beta_2)^T \in \mathbb{R}^4$.

Před samotným odvozováním ještě udělejme krátkou poznámku týkající se notace. Přestože v sekci 2.1 jsme používali u pro proměnnou v dopředné rovnici a d pro označení dat, u konkrétních modelů se budeme držet notace, která je typická pro tyto modely. Místo obecného u tedy budeme používat proměnné x, y pro Volterrův-Lotkův model, proměnné S, I, R pro SIR model apod. Příslušná data pak budeme značit, jako x_d, y_d atd., namísto obecného d .

Nyní přistoupíme k odvození adjungované rovnice a gradientu pro tento model. Pro zjednodušení popíšeme odvození podrobně pouze zde, pro SIR a Lorenzův model budou odvození zcela analogická.

Dosaď me model do obecně získané adjungované rovnice (2.10). Připomeňme, že používáme značení $u = (u_1, u_2) = (x, y)$ a $d = (d_1, d_2) = (x_d, y_d)$. Rozepsáním všech členů získáme

$$\frac{\partial \lambda^T}{\partial t} = \left(\frac{\partial \lambda_1}{\partial t}, \frac{\partial \lambda_2}{\partial t} \right), \quad (2.16)$$

$$-\lambda^T \frac{\partial f}{\partial u} = -(\lambda_1 \ \lambda_2) \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = -(\lambda_1 \ \lambda_2) \begin{pmatrix} \alpha_1 - \beta_1 y & -\beta_1 x \\ \beta_2 y & -\alpha_2 + \beta_2 x \end{pmatrix} \quad (2.17)$$

$$= -\left(\lambda_1(\alpha_1 - \beta_1 y) + \lambda_2 \beta_2 y, -\lambda_1 \beta_1 x + \lambda_2(-\alpha_2 + \beta_2 x) \right). \quad (2.18)$$

Rozepsáním po složkách získáme adjungovanou úlohu pro λ s podmínkou danou v konečném čase T tvaru

$$\begin{aligned} \frac{\partial \lambda_1}{\partial t} &= 2(x - x_d) - \lambda_1(\alpha_1 - \beta_1 y) - \lambda_2 \beta_2 y, \quad \text{na } [0, T] \\ \frac{\partial \lambda_2}{\partial t} &= 2(y - y_d) + \lambda_1 \beta_1 x - \lambda_2(-\alpha_2 + \beta_2 x), \\ \lambda_1(T) &= 0, \\ \lambda_2(T) &= 0. \end{aligned} \quad (2.19)$$

Dále můžeme přistoupit k odvození gradientu. Opět vyjdeme z obecně odvozeného tvaru (2.13). Obdobným rozepsáním derivací získáme

$$-\lambda^T \frac{\partial f}{\partial p} = -(\lambda_1 \ \lambda_2) \begin{pmatrix} \frac{\partial f_1}{\partial p_1} & \cdots & \frac{\partial f_1}{\partial p_4} \\ \frac{\partial f_2}{\partial p_1} & \cdots & \frac{\partial f_2}{\partial p_4} \end{pmatrix} = -(\lambda_1 \ \lambda_2) \begin{pmatrix} x & -xy & 0 & 0 \\ 0 & 0 & -x & xy \end{pmatrix} \quad (2.20)$$

$$= \left(-\lambda_1 x, \lambda_1 xy, \lambda_2 y, -\lambda_2 xy \right). \quad (2.21)$$

Dosažením do integrálu pak získáme konečný výsledek

$$\nabla J^*(p) = \int_0^T \left(-\lambda_1 x, \lambda_1 xy, \lambda_2 y, -\lambda_2 xy \right) dt. \quad (2.22)$$

2.3.2 SIR model

SIR model [4] a příbuzné modely tvoří soubor modelů používaných pro popis vývoje epidemií. Zde se budeme zabývat nejjednodušším takovým modelem. Populace je rozdělena na tři skupiny: S - náchylní k nakažení (susceptible), I - nakažení (infectious), R - uzdravení (recovered).

Dopředná rovnice je tvaru

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta IS}{N}, \quad \text{na } [0, T] \\ \frac{dI}{dt} &= \frac{\beta IS}{N} - \gamma I, \\ \frac{dR}{dt} &= \gamma I, \end{aligned} \quad (2.23)$$

kde $N = S(t) + I(t) + R(t)$; $\beta, \gamma > 0$ a $p = (\beta, \gamma) \in \mathbb{R}^2$. Parametr β udává tempo šíření infekce a parametr γ udává tempo přesunu jedinců ze skupiny infekčních do vyléčených.

Z tvaru systému je vidět, že v tomto případě uvažujeme celkovou populaci v čase konstantní neboť $\frac{dS}{dt}(t) + \frac{dI}{dt}(t) + \frac{dR}{dt}(t) = 0$, a tedy $S(t) + I(t) + R(t) = N = \text{const}$.

Dosadíme-li opět do obecně odvozených rovnic, získáme adjungovanou úlohu pro $\lambda^T = (\lambda_1, \lambda_2, \lambda_3)$ tvaru

$$\begin{aligned}\frac{\partial \lambda_1}{\partial t} &= 2(S - S_d) + \lambda_1 \frac{\beta I}{N} - \lambda_2 \frac{\beta I}{N}, \quad \text{na } [0, T] \\ \frac{\partial \lambda_2}{\partial t} &= 2(I - I_d) + \lambda_1 \frac{\beta S}{N} - \lambda_2 \left(\frac{\beta S}{N} - \gamma \right) - \lambda_3 \gamma, \\ \frac{\partial \lambda_3}{\partial t} &= 2(R - R_d), \\ \lambda_1(T) &= 0, \\ \lambda_2(T) &= 0, \\ \lambda_3(T) &= 0,\end{aligned}\tag{2.24}$$

a gradient se dále zjednoduší na tvar

$$\nabla J^*(p) = \int_0^T \left((\lambda_1 - \lambda_2) \frac{IS}{N}, (\lambda_2 - \lambda_3) I \right) dt.\tag{2.25}$$

2.3.3 Lorenzův model

Lorenzův model je klasický případ systému vykazující chaotické chování, který lze získat jako zjednodušenou verzi systému rovnic popisující proudění tekutin a konvekce v atmosféře. Přesným fyzikálním významem tohoto modelu se zde nebudeme zabývat, více lze najít v [3].

Dopředná rovnice má tvaru

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z,\end{aligned}\tag{2.26}$$

kde $\sigma, \rho, \beta > 0$ a $p = (\sigma, \rho, \beta)$.

Adjungovaná úloha pro $\lambda^T = (\lambda_1, \lambda_2, \lambda_3)$ má tvar

$$\begin{aligned}\frac{\partial \lambda_1}{\partial t} &= 2(x - x_d) + \lambda_1 \sigma - \lambda_2(\rho - z) - \lambda_3 y, \\ \frac{\partial \lambda_2}{\partial t} &= 2(y - y_d) - \lambda_1 \sigma + \lambda_2 - \lambda_3 x, \\ \frac{\partial \lambda_3}{\partial t} &= 2(z - z_d) + \lambda_2 x + \lambda_3 \beta, \\ \lambda_1(T) &= 0, \\ \lambda_2(T) &= 0, \\ \lambda_3(T) &= 0,\end{aligned}\tag{2.27}$$

a gradient má tvar

$$\nabla J^*(p) = \int_0^T \left(\lambda_1(x - y), -\lambda_2 x, \lambda_3 z \right) dt.\tag{2.28}$$

2.4 Numerické řešení

V této kapitole představíme numerické řešení jednotlivých dopředných a adjungovaných rovnic, gradientu a ztrátové funkce. Nejprve si připravíme diskretizaci časového intervalu a časové derivace.

- **Diskretizace časového intervalu:**

Zaved' me síť časových kroků jako

$$\{k\tau \mid \tau > 0, k = 0, \dots, M\},$$

kde $\tau > 0$ je časový krok, $M + 1$ je počet časových hladin a platí $M = \lfloor \frac{T}{\tau} \rfloor$.

Dále pro veličinu $z = z(t)$ zaved' me značení

$$z^k := z(k\tau), \quad k = 0, \dots, M.$$

Pro větší přehlednost zápisu budeme také na některých místech používat značení

$$z|_k := z(k\tau), \quad k = 0, \dots, M.$$

- **Nahrazení prvních derivací:**

K aproximaci prvních derivací použijeme náhradu pomocí dopředné diference. Bud' $z = z(t)$ veličina jejíž první derivaci chceme aproximovat. Diferenční náhrada pak vypadá jako

$$\frac{dz}{dt}(k\tau) \approx \frac{z^{k+1} - z^k}{\tau}, \quad k = 0, \dots, M - 1. \quad (2.29)$$

Nyní můžeme přistoupit k sestavení diferenčních schémat.

2.4.1 Dopředná rovnice

Mějme spojitou počáteční úlohu (2.1)

$$\frac{du}{dt}(t, p) = f(t, p, u(t, p)), \quad \text{na } [0, T] \quad (2.30)$$

$$u(0, p) = u_0. \quad (2.31)$$

Náhradou první derivace výrazem (2.29) získáme soustavu algebraických rovnic tvaru

$$u^{k+1} = u^k + \tau f^k, \quad k = 0, \dots, M - 1, \quad (2.32)$$

s počáteční podmínkou

$$u^0 = u(0). \quad (2.33)$$

Tento tvar odpovídá Eulerově metodě řešení diferenciálních rovnic. Pro zlepšení numerického řešení se často používají Rungeovy-Kuttovy metody, které zde však z podstaty problému nejsou vhodné, jelikož při výpočtu adjungované rovnice používáme výsledky již vypočtené dopředné rovnice, kterou však máme pouze v určitých časových hladinách. Naproti tomu můžeme použít vícekrokové Adamsovy-Bashforthovy formule, které používají více předchozích časových hladin, čímž dosahují lepší přesnosti i pro větší hodnoty časového kroku τ .

V této práci použijeme Adamsovu-Bashforthovu formuli druhého řádu [7], která má v našem zápisu tvar

$$u^{k+2} = u^{k+1} + \frac{\tau}{2} \left(3f^{k+1} - f^k \right), \quad k = 0, \dots, M-2, \quad (2.34)$$

s počátečními podmínkami

$$u^0 = u(0), \quad (2.35)$$

$$u^1 = u^0 + \tau f(0). \quad (2.36)$$

2.4.2 Adjungovaná rovnice

Analogicky získáme diferenční schéma pro úlohu (2.10). Opět představíme schéma pro Eulerovu a Adamsovu-Bashforthovu metodu. Eulerova metoda má tvar

$$\lambda^{T,k-1} = \lambda^{T,k} + \tau \left(2(u^k - d^k) - \lambda^{T,k} \frac{\partial f}{\partial u} \Big| ^k \right), \quad k = M, \dots, 1, \quad (2.37)$$

s podmínkou danou v konečném čase

$$\lambda^{T,M} = 0. \quad (2.38)$$

Adamsova-Bashforthova metoda má tvar

$$\lambda^{T,k-2} = \lambda^{T,k-1} + \frac{\tau}{2} \left(3 \left(2(u^{k-1} - d^{k-1}) - \lambda^{T,k-1} \frac{\partial f}{\partial u} \Big| ^{k-1} \right) - 2(u^k - d^k) + \lambda^{T,k} \frac{\partial f}{\partial u} \Big| ^k \right), \quad k = M, \dots, 2, \quad (2.39)$$

s podmínkami zadanými v posledních dvou časových hladinách

$$\lambda^{T,M} = 0, \quad (2.40)$$

$$\lambda^{T,M-1} = \lambda^{T,M} + \tau \left(2(u^M - d^M) - \lambda^{T,M} \frac{\partial f}{\partial u} \Big| ^M \right). \quad (2.41)$$

2.4.3 Gradient

Pro numerickou integraci použijeme lichoběžníkovou metodu [7]. Touto metodou můžeme aproximovat gradient (2.13)

$$\nabla J^*(p) = \int_0^T \left(-\lambda^T \frac{\partial f}{\partial p} \right) dt,$$

jako

$$\nabla J^*(p) \approx \sum_{k=0}^{M-1} \left(-\lambda^{T,k} \frac{\partial f}{\partial p} \Big| ^k - \lambda^{T,k+1} \frac{\partial f}{\partial p} \Big| ^{k+1} \right) \frac{\tau}{2}. \quad (2.42)$$

2.4.4 Ztrátová funkce

Nakonec přidejme postup pro výpočet ztrátové funkce (2.4) tvaru

$$J(p) = \int_0^T (u(t, p) - d(t))^2 dt,$$

jako

$$J(p) \approx \sum_{k=0}^{M-1} \left((u^k - d^k)^2 + (u^{k+1} - v^{k+1})^2 \right) \frac{\tau}{2}. \quad (2.43)$$

2.5 Implementace

V této části stručně shrneme algoritmy používané k odhadu parametrů a jejich implementaci v programovacím jazyce C++. Jak již bylo zmíněno výše, naším cílem je odhad parametrů modelů z dat za pomoci výpočtu gradientu a následné aktualizace parametrů za pomoci algoritmu RMSprop. Výše jsme naznačili dva možné výpočty gradientu, buď pomocí řešení adjungované rovnice, a nebo přímým výpočtem pomocí aproximace metodou konečných diferencí. Než přistoupíme k výpočetní studii, shrňme v pseudokódu oba přístupy.

Algoritmus 2: Výpočet gradientu adjungovanou rovnicí.

Vstup: Data $d(t)$

Vstup: Parametry α, ρ algoritmu RMSprop

Vstup: Počáteční parametry p_0

Vstup: Prahová hodnota

- 1: **while** Hodnota ztrátové funkce > prahová hodnota **do**
 - 2: Vyřeš dopřednou rovnici (2.1) pro u
 - 3: Vyřeš adjungovanou rovnici (2.10) pro λ^T
 - 4: Vypočti gradient (2.13) a normalizuj ho
 - 5: Proveď krok v RMSprop algoritmu (1.7)
 - 6: Vypočti hodnotu ztrátové funkce podle (2.4)
 - 7: **end while**
-

Algoritmus 3: Výpočet gradientu metodou konečných diferencí.

Vstup: Data $d(t)$

Vstup: Parametry α, ρ algoritmu RMSprop

Vstup: Počáteční parametry p_0

Vstup: Prahová hodnota

- 1: **while** Hodnota ztrátové funkce > prahová hodnota **do**
 - 2: Vypočti gradient pomocí (2.14) a normalizuj ho
 - 3: Proveď krok v RMSprop algoritmu (1.7)
 - 4: Vypočti hodnotu ztrátové funkce podle (2.4)
 - 5: **end while**
-

Vhodné nastavení vstupních údajů pro tyto algoritmy závisí na konkrétní úloze a často se velmi liší. Přesto shrňme alespoň v obecných rysech, jak budeme tyto vstupní data nastavovat.

Data $d(t)$ v našem případě získáme z řešení dopředné rovnice s předem určenými parametry p_{data} .

Vhodné hodnoty parametrů α a ρ v algoritmu RMSprop závisí na konkrétní úloze, a jak uvidíme dále, často je nutné vyzkoušet více kombinací. Jako univerzální počáteční odhad lze použít například hodnoty $\alpha = 10^{-4}$ a $\rho = 0.99$.

Vhodné zvolení počátečního odhadu parametrů p_0 se ukazuje jako zásadní a špatný počáteční odhad často končí divergencí programu. Současně zřejmě neexistuje univerzální postup nebo heuristika na to, jak počáteční parametry volit. Z pozorování vyplývá, že nejlepší šancí je vyjít z apriorní znalosti modelu a rozumných hodnot jeho parametrů. Další možnost nabízí různé metody prohledávání prostoru parametrů pro nalezení vhodného počátečního odhadu. Z jednodušších takových metod zmiňme například Grid Search nebo Random Search [2].

2.6 Výpočetní studie

V této sekci postupně pro každý z modelů předvedeme výpočty dopředné rovnice, ilustrované obrázky a následně přistoupíme k samotnému odhadu parametrů pomocí obou metod. U všech modelů budeme postupovat tak, že zvolíme pevně dané parametry p_{data} , které se později budeme pokoušet odhadnout z různých počátečních odhadů p_0 . Vyřešením dopředné rovnice s těmito parametry také současně získáme data $d(t)$.

Poznamenejme, že všechny simulace provedené v této práci byly spuštěny na notebooku s 8 jádrovým procesorem Intel Core i5-1135G7.

Nyní máme vše připravené a můžeme přistoupit k samotným modelům.

• Volterrův-Lotkův model

Parametry dopředné rovnice, a tedy i dat $d(t)$, zvolme jako

$$p_{\text{data}} = (\alpha_1, \beta_1, \alpha_2, \beta_2)_{\text{data}} = (0.5471, 0.281, 0.8439, 0.0266),$$

kteřé vedou na periodické chování systému.

Jako časový interval zvolíme $[0, 20]$ a počáteční podmínky nastavíme jako $x(0) = 30$ a $y(0) = 4$. Řešíme tedy počáteční úlohu (2.15) tvaru

$$\frac{dx}{dt} = \alpha_1 x - \beta_1 xy, \quad \text{na } [0, 20] \quad (2.44)$$

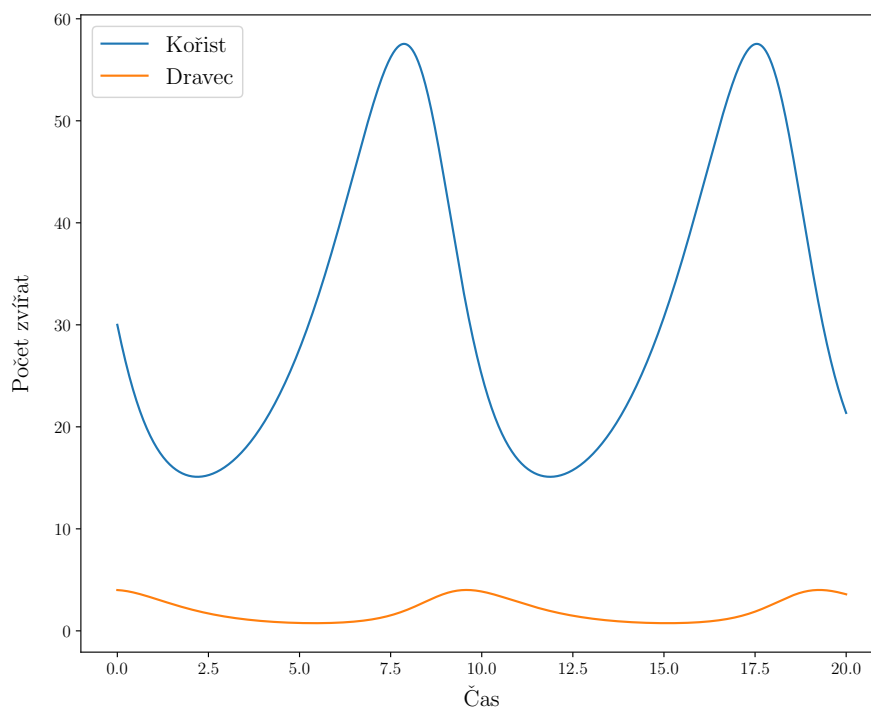
$$\frac{dy}{dt} = -\alpha_2 y + \beta_2 xy, \quad (2.45)$$

$$x(0) = 30, \quad (2.46)$$

$$y(0) = 4, \quad (2.47)$$

pro $(\alpha_1, \beta_1, \alpha_2, \beta_2) = (\alpha_1, \beta_1, \alpha_2, \beta_2)_{\text{data}}$. K numerickému řešení zvolíme Adamsovu-Bashforthovu metodu a časový krok zvolíme, jako $\tau = 10^{-2}$.

Výsledek simulace dopředné rovnice s tímto nastavením můžeme vidět na obrázku 2.1.



Obrázek 2.1: Volterrův-Lotkův model pro $p = (0.5471, 0.281, 0.8439, 0.0266)$.

Nyní můžeme přistoupit k odhadu parametrů. Jak už bylo řečeno výše, data $d(t)$ jsme získali vyřešením dopředné rovnice s těmito parametry. Jak se však ukazuje, odhad parametrů je nutné provést na kratším časovém intervalu, jelikož na delším intervalu mají oba algoritmy problémy s konvergencí. Tím budeme náš model fitovat pouze na podmnožinu získaných dat, nicméně se ukazuje, že pro odhad parametrů to není problém. Jak správně zvolit časový interval, ze kterého budeme chtít odhadnout parametry, opět není jednoduché a podobně, jako u počáteční volby parametrů je zřejmě nutné vyzkoušet několik intervalů a vybrat ten, který má nejlepší výsledky.

Pro oba algoritmy dále požadujeme parametry α a ρ pro RMSprop, které zvolíme stejným způsobem, jak bylo naznačeno v kapitole o implementaci.

Počáteční parametry zvolíme několika různými způsoby, abychom ukázali závislost algoritmu na jejich vhodném odhadu a také rozdíl ve funkcionalitě obou algoritmů. Jako sadu testovacích počátečních parametrů zvolme kombinace v tabulce 2.1.

Tabulka 2.1: Počáteční odhady parametrů.

Hodnoty parametrů
(0.05, 0.05, 0.05, 0.05)
(0.1, 0.1, 0.1, 0.1)
(0.2, 0.2, 0.2, 0.2)
(0.5, 0.5, 0.5, 0.5)

Tyto hodnoty ilustrují situaci, kdy nemáme nějaké apriorní znalosti o typických hodnotách parametrů. Jak už bylo zmíněno výše, jednou z možností, jak volit lepší počáteční odhady by mohla být lepší teoretická znalost modelu, která by například mohla předem vyloučit některé kombinace. Protože tyto hodnoty nepovedou k příliš dobrým výsledkům, ukážeme dále, jak zdánlivě malou změnou v počátečních odhadech získat o mnoho lepší výsledky.

1. Výpočet gradientu pomocí adjungované rovnice

Nyní budeme postupovat podle algoritmu 2. Jelikož všechny vstupy pro algoritmus, tedy data $d(t)$, parametry α a ρ a počáteční parametry máme připravené, můžeme přejít k simulaci. Jako časový interval použijeme $[0, 5]$ a časový krok zmenšíme na $\tau = 10^{-4}$. K normalizaci gradientu na počátku optimalizace použijeme standardizaci (1.9).

Výsledky algoritmu jsou shrnuty v tabulce 2.2. Druhý a třetí sloupec obsahují buď hodnotu ztrátové funkce a výsledné hodnoty naučených parametrů nebo "-" v případě, že algoritmus nedoběhl.

Tabulka 2.2: Odhad parametrů pomocí adjungované rovnice.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.05, 0.05, 0.05, 0.05)	2.4×10^{-4}	(0.54775, 0.28123, 0.84099, 0.02645)
(0.1, 0.1, 0.1, 0.1)	-	-
(0.2, 0.2, 0.2, 0.2)	-	-
(0.5, 0.5, 0.5, 0.5)	-	-

Jak vidíme z tabulky algoritmus na těchto počátečních odhadech příliš dobře nezafungoval a většinou selhal. Jako problém se ukazuje čtvrtý parametr β_2 , který má menší velikost než ostatní tři parametry. Zvolíme-li čtvrtý parametr jako velmi malý nebo nulový, stabilita algoritmu se zlepší viz. tabulka 2.3.

Tabulka 2.3: Odhad parametrů s upravenou volbou parametru.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.05, 0.05, 0.05, 0.01)	7.8×10^{-5}	(0.54898, 0.28165, 0.83783, 0.02631)
(0.1, 0.1, 0.1, 0.01)	8.6×10^{-5}	(0.54901, 0.28167, 0.83756, 0.0263)
(0.2, 0.2, 0.2, 0.01)	5.4×10^{-5}	(0.54865, 0.28151, 0.83904, 0.02637)
(0.5, 0.5, 0.5, 0.01)	3.8×10^{-4}	(0.54521, 0.28032, 0.85041, 0.02692)

Jak je vidět v případě algoritmu 2, závisí jeho stabilita zásadně na počátečních parametrech p_0 . Důvodem je fakt, že jak jsme zmiňovali, adjungovaná rovnice je obyčejná lineární

diferenciální rovnice, a tedy její řešení umíme explicitně zapsat jako lineární kombinaci exponenciálních funkcí. Ukazuje se, že pokud se během optimalizace vlastní čísla soustavy tvořící exponenty nakombinují takovým způsobem, že alespoň jedno z nich vyjde jako relativně velké pozitivní číslo, pak hodnoty adjungované rovnice vyjdou v absolutní hodnotě jako velmi velké, což začne způsobovat numerickou nestabilitu. Protože algoritmus 2 běží v cyklu, tento jev se opakováním typicky ještě zesílí, což způsobí divergenci a pád programu. Toto je také jeden z důvodů, proč je vhodnější volit kratší časové intervaly, protože na nich má adjungovaná rovnice menší šanci nabýt velkých hodnot.

CPU časy výpočtu se pohybovaly v závislosti na požadavku na přesnost v rozmezí mezi 30 sekundami a nižšími minutami.

2. Výpočet gradientu metodou konečných diferencí

Nyní budeme postupovat podle algoritmu 3. Parametr τ zvětšíme na $\tau = 10^{-3}$ a ostatní nastavení ponecháme stejné jako u předchozí části. Výsledky jsou shrnuty v tabulce 2.4.

Tabulka 2.4: Odhad parametrů pomocí konečných diferencí.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.05, 0.05, 0.05, 0.05)	2.8×10^{-5}	(0.5472, 0.28103, 0.83913, 0.0263)
(0.1, 0.1, 0.1, 0.1)	9.8×10^{-5}	(0.5468, 0.28061, 0.83847, 0.0263)
(0.2, 0.2, 0.2, 0.2)	4.5×10^{-4}	(0.54602, 0.27978, 0.8337, 0.0262)
(0.5, 0.5, 0.5, 0.5)	4.1×10^{-3}	(0.54365, 0.27723, 0.83303, 0.02586)

Jak je vidět z tabulky, výpočet pomocí konečných diferencí se zdá být stabilnější vzhledem k počátečnímu odhadu, jelikož stabilitu výpočtu nemůže ohrozit adjungovaná rovnice. Současně jsme parametry odhadli i s větším časovým krokem a CPU časy se proto pohybovaly shodně kolem 30 sekund.

• SIR model

V této části zvolme několik různých parametrů dat s různou hodnotou parametru β simulující různé nakažlivé nemoci. Dvojice parametrů, které se budeme snažit odhadnout jsou shrnuty v tabulce 2.5.

Tabulka 2.5: Parametry dat.

Hodnoty parametrů
(0.6, 0.4)
(0.8, 0.4)
(1.2, 0.4)
(2.0, 0.4)

Nejprve si připravíme data $d(t)$, která získáme řešením dopředné rovnice s výše zmíněnými parametry a příslušnými počátečními podmínkami. Úlohu řešíme na časovém intervalu $[0, 50]$. Celá počáteční úloha pak má tvar

$$\frac{dS}{dt} = -\frac{\beta IS}{N}, \quad \text{na } [0, 50] \quad (2.48)$$

$$\frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I, \quad (2.49)$$

$$\frac{dR}{dt} = \gamma I, \quad (2.50)$$

$$S(0) = 990, \quad (2.51)$$

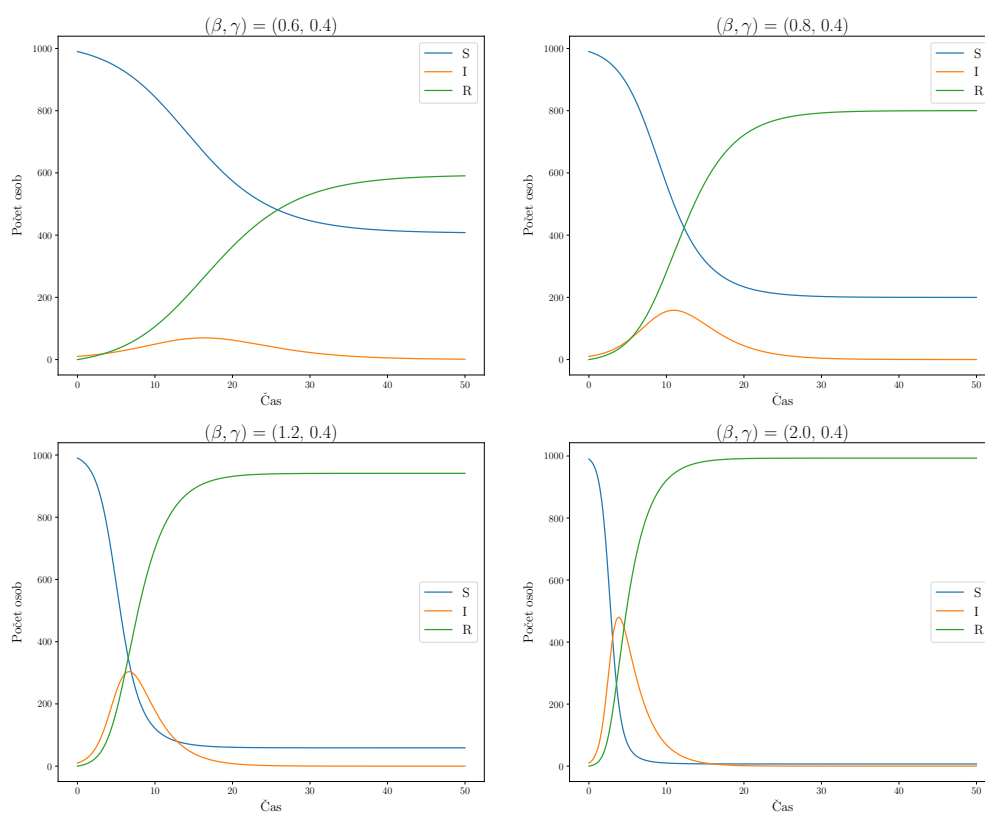
$$I(0) = 10, \quad (2.52)$$

$$R(0) = 0, \quad (2.53)$$

$$(2.54)$$

pro parametry (β, γ) z tabulky 2.5. K numerickému řešení použijeme Adamsovu-Bashforthovu metodu a časový krok zvolíme jako $\tau = 10^{-2}$.

Výsledky simulace jsou na obrázku 2.2.



Obrázek 2.2: SIR model pro různé parametry.

Nyní přistoupíme k odhadu parametrů. Hodnoty parametrů α a ρ pro RMSprop opět ponechme v základním nastavení. K normalizaci gradientu použijeme obyčejnou normalizaci.

1. Výpočet gradientu pomocí adjungované rovnice

Zvolme počáteční odhad parametrů, jako $p_0 = (0.5, 0.5)$. Konečný čas T zvolme, jako $T = 5$. Časový krok i numerickou metodu ponechme stejné. Výsledky jsou shrnuty v tabulce 2.6.

Tabulka 2.6: Odhad parametrů pomocí adjungované rovnice.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.6, 0.4)	2.8×10^{-10}	(0.6, 0.4)
(0.8, 0.4)	7.7×10^{-7}	(0.79999, 0.0.40008)
(1.2, 0.4)	4.1×10^{-5}	(1.19999, 0.4)
(2.0, 0.4)	1.3×10^{-10}	(2.0, 0.4)

2. Výpočet gradientu metodou konečných diferencí

Dále provedeme odhad rovněž pomocí konečných diferencí. Všechny nastavení ponechme stejné jako u odhadu pomocí adjungované rovnice. Výsledky jsou shrnuty v tabulce 2.7.

Tabulka 2.7: Odhad parametrů pomocí konečných diferencí.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.6, 0.4)	1.3×10^{-5}	(0.6, 0.39999)
(0.8, 0.4)	4.2×10^{-5}	(0.79999, 0.0.40002)
(1.2, 0.4)	3.6×10^{-5}	(1.19999, 0.4)
(2.0, 0.4)	9.7×10^{-6}	(1.99999, 0.4)

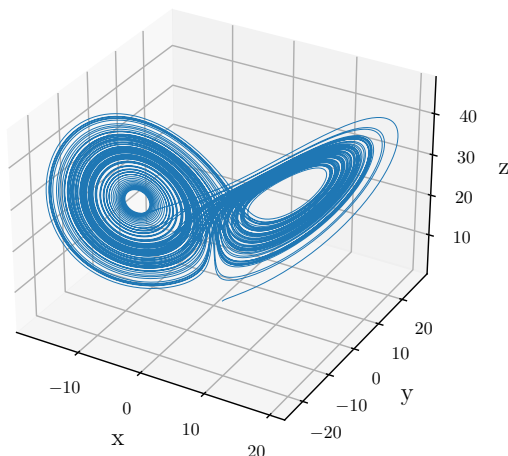
CPU časy obou metod se pohybovaly shodně kolem několika sekund.

• Lorenzův model

Nejprve přidáme výpočty dopředné rovnice. Jako ilustrační příklad parametrů p_{data} zvolíme známou kombinaci vedoucí na chaotické chování, a to $p_{\text{data}} = (10, 28, \frac{8}{3})$. Jako časový interval zvolme $[0, 100]$. Počáteční podmínky zvolme jako $x(0) = y(0) = z(0) = 1$. Celá počáteční úloha má pak tvar

$$\begin{aligned}
 \frac{dx}{dt} &= \sigma(y - x), \quad t \in [0, 100] \\
 \frac{dy}{dt} &= x(\rho - z) - y, \\
 \frac{dz}{dt} &= xy - \beta z, \\
 x(0) &= 1, \\
 y(0) &= 1, \\
 z(0) &= 1,
 \end{aligned} \tag{2.55}$$

pro $(\sigma, \rho, \beta) = p_{\text{data}}$. K řešení použijeme Adamsovu-Bashforthovu metodu a časový krok zvolme jako $\tau = 10^{-3}$. Výsledek je zachycený na obrázku 2.3.



Obrázek 2.3: Lorenzův atraktor.

Nyní můžeme přistoupit k odhadu parametrů. Vzhledem k původu Lorenzova modelu zafixujeme hodnoty parametrů σ a β a zkusíme odhadnout parametr ρ pro různé hodnoty. Z teorie [3] plyne, že existuje kritická hodnota parametru $\rho \approx 24.74$. Pokud zvolíme hodnoty parametru $\rho > 24.74$, systém vykazuje chaotické chování. Pro jednoduchost zvolme $\rho = 28$ a pokusme se tento parametr odhadnout.

1. Výpočet gradientu pomocí adjungované rovnice

Pomocí adjungované rovnice se parametry nepodařilo odhadnout.

2. Výpočet gradientu metodou konečných diferencí

Časový interval zvolme jako $[0, 5]$. Ostatní nastavení ponechme stejná jako u simulace dopředné rovnice. Parametry algoritmu RMSprop ponechme v základním nastavení a k normalizaci gradientu použijme obyčejnou normalizaci. Výsledky pro různé počáteční hodnoty parametru ρ jsou shrnuty v tabulce 2.8.

Tabulka 2.8: Odhad parametrů pomocí konečných diferencí.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(10.0, 20.0, 2.66)	1.3×10^{-3}	(10.0197, 27.999, 2.66035)
(10.0, 25.0, 2.66)	1.9×10^{-4}	(10.0074, 27.9996, 2.66014)
(10.0, 30.0, 2.66)	8.1×10^{-5}	(9.995, 28.0002, 2.65982)

2.7 Diskuze výsledků

V této sekci krátce porovnáme oba algoritmy a následně shrneme jejich funkčnost pro jednotlivé modely.

Z výsledků v sekci 2.6 je vidět, že z obou algoritmů lépe fungoval odhad pomocí metody konečných diferencí, protože pro Lorenzův model se odhad pomocí adjungované rovnice vůbec nepodařil. Další výhodou metody konečných diferencí je jednodušší nastavení algoritmu, kdy parametry simulace šlo typicky ponechat ve stejném nastavení, které jsme používali pro simulaci dopředné rovnice. Naproti tomu pro výpočet gradientu pomocí adjungované rovnice bylo správné nastavení těžší, protože často bylo nutné nejprve najít časový interval, na kterém adjungovaná rovnice bude nabývat rozumných hodnot. Teoretická výhoda adjungované rovnice, kterou jsme debatovali v předchozích sekcích tj. konstantní časová náročnost v závislosti na počtu parametrů je pro tuto úlohu zanedbatelná, neboť největší počet parametrů (4 pro Volterrův-Lotkův model) nezpůsobil žádný významný časový nárůst.

Kapitola 3

Odhad parametrů pro Grayův-Scottův model

V této kapitole zopakujeme podobný postup, který jsme předvedli v minulé kapitole o obyčejných diferenciálních rovnicích s tím rozdílem, že nyní odvodíme adjungovanou rovnici pouze pro Grayův-Scottův model [6].

3.1 Grayův-Scottův model

Jedná se o soustavu dvou parciálních diferenciálních rovnic parabolického typu popisující časový vývoj koncentrací dvou chemických sloučenin u a v , které spolu interagují pomocí difúze a chemické reakce. Grayův-Scottův model popisuje chemickou reakci tvaru



kde P je interní produkt. Výsledné rovnice mají tvar

$$\frac{\partial u}{\partial t} = D_u \Delta u - uv^2 + f(1 - u), \quad (3.3)$$

$$\frac{\partial v}{\partial t} = D_v \Delta v + uv^2 - (f + k)v, \quad (3.4)$$

kde $D_u, D_v > 0$ jsou difúzní koeficienty a parametry $f > 0$ (feed), $k > 0$ (kill) popisují intenzitu chemické reakce. V tomto případě zavádíme $p := (f, k)$. Pro následující odvození uvažujeme funkce $u = u(x, t, p)$ a $v = v(x, t, p)$. Dále zaved' me označení pro polynomy vyskytující se v rovnicích, které později využijeme

$$F(u, v) := -uv^2 + f(1 - u), \quad (3.5)$$

$$G(u, v) := uv^2 - (f + k)v. \quad (3.6)$$

V závislosti na kombinaci koeficientů f a k můžeme při řešení Grayova-Scottova modelu pozorovat různé vzory. Několik ilustračních případů pro různé kombinace parametrů bude v sekci 3.5.

3.2 Odvození adjungované rovnice

Nyní můžeme přejít k odvození adjungované rovnice pro Grayův-Scottův model. Poznamenejme, že v následujícím odvození nebudeme pro přehlednost explicitně vyznačovat, v jakých bodech funkce u a v vyčíslujeme, pokud to bude z kontextu jasné. Ztrátová funkce $J(p)$ má tvar

$$J(p) = \frac{1}{2} \int_0^T \int_{\Omega} \left((u - u_d)^2 + (v - v_d)^2 \right) dx dt, \quad (3.7)$$

kde $u_d = u_d(x, t)$ a $v_d = v_d(x, t)$ jsou naměřená data pocházející z Grayova-Scottova modelu a $\Omega \subset \mathbb{R}^2$ je omezená oblast, na níž úlohu řešíme.

Optimalizaci provádíme za platnosti dopředné rovnice s nulovými Dirichletovými okrajovými podmínkami a danými počátečními podmínkami $u_0 = u_0(x)$ a $v_0 = v_0(x)$

$$\frac{\partial u}{\partial t} = D_u \Delta u - uv^2 + f(1 - u) \quad \text{na } \Omega \times [0, T], \quad (3.8)$$

$$\frac{\partial v}{\partial t} = D_v \Delta v + uv^2 - (f + k)v \quad \text{na } \Omega \times [0, T], \quad (3.9)$$

$$u = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.10)$$

$$v = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.11)$$

$$u(x, 0) = u_0(x) \quad \text{na } \Omega, \quad (3.12)$$

$$v(x, 0) = v_0(x) \quad \text{na } \Omega. \quad (3.13)$$

Přesný tvar počátečních podmínek u_0 a v_0 pro náš případ upřesníme v sekci 3.5. Řešíme tedy úlohu na hledání vázaného extrému s vazbou danou soustavou parciálních diferenciálních rovnic, k čemuž opět použijeme metodu Lagrangeových multiplikátorů. Zaved' me multiplikátory $\lambda_1, \lambda_2, \lambda_3$ a μ_1, μ_2, μ_3 , kde

$$\lambda_1 = \lambda_1(x, t), \quad \mu_1 = \mu_1(x, t) \quad \text{na } \Omega \times [0, T], \quad (3.14)$$

$$\lambda_2 = \lambda_2(x, t), \quad \mu_2 = \mu_2(x, t) \quad \text{na } \partial\Omega \times [0, T], \quad (3.15)$$

$$\lambda_3 = \lambda_3(x), \quad \mu_3 = \mu_3(x) \quad \text{na } \Omega. \quad (3.16)$$

Rozšířenou ztrátovou funkci $J^*(p)$ potom můžeme zapsat ve tvaru

$$\begin{aligned} J^*(p) &= \frac{1}{2} \int_0^T \int_{\Omega} \left((u - u_d)^2 + (v - v_d)^2 \right) dx dt \\ &+ \int_0^T \int_{\Omega} \lambda_1 \left(\frac{\partial u}{\partial t} - D_u \Delta u + uv^2 - f(1 - u) \right) dx dt \\ &+ \int_0^T \int_{\Omega} \mu_1 \left(\frac{\partial v}{\partial t} - D_v \Delta v - uv^2 + (f + k)v \right) dx dt \\ &+ \int_0^T \int_{\partial\Omega} \left(\lambda_2 u + \mu_2 v \right) dS dt \\ &+ \int_{\Omega} \left(\lambda_3 (u(x, 0) - u_0(x)) + \mu_3 (v(x, 0) - v_0(x)) \right) dx. \end{aligned}$$

Derivací podle vektoru parametrů p získáme (pro přehlednost nahradíme pravé strany funkcemi F a G)

$$\begin{aligned} \frac{dJ^*}{dp} &= \int_0^T \int_{\Omega} \left((u - u_d) \frac{\partial u}{\partial p} + (v - v_d) \frac{\partial v}{\partial p} \right) dx dt \\ &+ \int_0^T \int_{\Omega} \left(\lambda_1 \frac{\partial}{\partial p} \left(\frac{\partial u}{\partial t} \right) - D_u \frac{\partial}{\partial p} \Delta u - \frac{\partial F}{\partial p} \right) dx dt \\ &+ \int_0^T \int_{\Omega} \left(\mu_1 \frac{\partial}{\partial p} \left(\frac{\partial v}{\partial t} \right) - D_v \frac{\partial}{\partial p} \Delta v - \frac{\partial G}{\partial p} \right) dx dt \\ &+ \int_0^T \int_{\partial\Omega} \left(\lambda_2 \frac{\partial u}{\partial p} + \mu_2 \frac{\partial v}{\partial p} \right) dS dt \\ &+ \int_{\Omega} \left(\lambda_3 \left(\frac{\partial u}{\partial p}(x, 0) - \frac{\partial u_0}{\partial p}(x) \right) + \mu_3 \left(\frac{\partial v}{\partial p}(x, 0) - \frac{\partial v_0}{\partial p}(x) \right) \right) dx. \end{aligned}$$

Zvlášť odvodíme derivace pro funkce F a G

$$\frac{\partial F}{\partial p} = \frac{\partial}{\partial p} \left(-uw^2 + f(1-u) \right) = -v^2 \frac{\partial u}{\partial p} - 2uw \frac{\partial v}{\partial p} + (1, 0) - (1, 0)u - f \frac{\partial u}{\partial p} = -(v^2 + f) \frac{\partial u}{\partial p} - 2uw \frac{\partial v}{\partial p} + (1 - u, 0),$$

$$\frac{\partial G}{\partial p} = \frac{\partial}{\partial p} \left(uw^2 - (f+k)v \right) = v^2 \frac{\partial u}{\partial p} + 2uw \frac{\partial v}{\partial p} - (1, 0)v - f \frac{\partial v}{\partial p} - (0, 1)v - k \frac{\partial v}{\partial p} = v^2 \frac{\partial u}{\partial p} + (2uw - f - k) \frac{\partial v}{\partial p} - (v, v).$$

Obdobně jako v předchozí kapitole upravíme časové derivace pomocí integrace per partes

$$\int_0^T \int_{\Omega} \lambda_1 \frac{\partial}{\partial p} \left(\frac{\partial u}{\partial t} \right) = \int_{\Omega} \left(\lambda_1 \frac{\partial u}{\partial p}(T) - \lambda_1 \frac{\partial u}{\partial p}(0) \right) dx - \int_0^T \int_{\Omega} \frac{\partial \lambda_1}{\partial t} \frac{\partial u}{\partial p} dx dt, \quad (3.17)$$

$$\int_0^T \int_{\Omega} \mu_1 \frac{\partial}{\partial p} \left(\frac{\partial v}{\partial t} \right) = \int_{\Omega} \left(\mu_1 \frac{\partial v}{\partial p}(T) - \mu_1 \frac{\partial v}{\partial p}(0) \right) dx - \int_0^T \int_{\Omega} \frac{\partial \mu_1}{\partial t} \frac{\partial v}{\partial p} dx dt. \quad (3.18)$$

Dále s dvojným použitím první Greenovy identity [10] můžeme upravit integraci přes výrazy obsahující Laplaceův operátor

$$-D_u \int_0^T \int_{\Omega} \lambda_1 \Delta \frac{\partial u}{\partial p} dx dt = -D_u \int_0^T \int_{\partial\Omega} \lambda_1 \frac{\partial}{\partial n} \frac{\partial u}{\partial p} dS dt + D_u \int_0^T \int_{\Omega} \nabla \lambda_1 \cdot \nabla \frac{\partial u}{\partial p} dx dt \quad (3.19)$$

$$= -D_u \int_0^T \int_{\Omega} \Delta \lambda_1 \frac{\partial u}{\partial p} dx dt + D_u \int_0^T \int_{\partial\Omega} \left(\frac{\partial \lambda_1}{\partial n} \frac{\partial u}{\partial p} - \lambda_1 \frac{\partial}{\partial p} \frac{\partial u}{\partial n} \right) dS dt, \quad (3.20)$$

$$-D_v \int_0^T \int_{\Omega} \mu_1 \Delta \frac{\partial v}{\partial p} dx dt = -D_v \int_0^T \int_{\partial\Omega} \mu_1 \frac{\partial}{\partial n} \frac{\partial v}{\partial p} dS dt + D_v \int_0^T \int_{\Omega} \nabla \mu_1 \cdot \nabla \frac{\partial v}{\partial p} dx dt \quad (3.21)$$

$$= -D_v \int_0^T \int_{\Omega} \Delta \mu_1 \frac{\partial v}{\partial p} dx dt + D_v \int_0^T \int_{\partial\Omega} \left(\frac{\partial \mu_1}{\partial n} \frac{\partial v}{\partial p} - \mu_1 \frac{\partial}{\partial p} \frac{\partial v}{\partial n} \right) dS dt. \quad (3.22)$$

Dosazením výrazů zpět dostaneme rozšířenou ztrátovou funkci ve tvaru

$$\begin{aligned}
\frac{dJ^*}{dp} = & \int_0^T \int_{\Omega} (u - u_d) \frac{\partial u}{\partial p} \, dx \, dt \\
& + \int_{\Omega} \lambda_1 \left(\frac{\partial u}{\partial p}(x, T) - \frac{\partial u}{\partial p}(x, 0) \right) dx \\
& - \int_0^T \int_{\Omega} \left(\frac{\partial \lambda_1}{\partial t} + D_u \Delta \lambda_1 \right) \frac{\partial u}{\partial p} \, dx \, dt \\
& + \int_0^T \int_{\partial \Omega} D_u \left(\frac{\partial \lambda_1}{\partial n} \frac{\partial u}{\partial p} - \lambda_1 \frac{\partial}{\partial p} \frac{\partial u}{\partial n} \right) dS \, dt \\
& - \int_0^T \int_{\Omega} \lambda_1 \left(-(v^2 + f) \frac{\partial u}{\partial p} - 2uv \frac{\partial v}{\partial p} + (1 - u, 0) \right) dx \, dt \\
& + \int_0^T \int_{\partial \Omega} \lambda_2 \frac{\partial u}{\partial p} \, dS \, dt \\
& + \int_{\Omega} \lambda_3 \left(\frac{\partial u}{\partial p}(x, 0) - \frac{\partial u_0}{\partial p}(x, 0) \right) dx \\
& + \int_0^T \int_{\Omega} (v - v_d) \frac{\partial v}{\partial p} \, dx \, dt \\
& + \int_{\Omega} \mu_1 \left(\frac{\partial v}{\partial p}(x, T) - \frac{\partial v}{\partial p}(x, 0) \right) dx \\
& - \int_0^T \int_{\Omega} \left(\frac{\partial \mu_1}{\partial t} + D_v \Delta \mu_1 \right) \frac{\partial v}{\partial p} \, dx \, dt \\
& + \int_0^T \int_{\partial \Omega} D_v \left(\frac{\partial \mu_1}{\partial n} \frac{\partial v}{\partial p} - \mu_1 \frac{\partial}{\partial p} \frac{\partial v}{\partial n} \right) dS \, dt \\
& - \int_0^T \int_{\Omega} \mu_1 \left(v^2 \frac{\partial u}{\partial p} + (2uv - f - k) \frac{\partial v}{\partial p} - (v, v) \right) dx \, dt \\
& + \int_0^T \int_{\partial \Omega} \mu_2 \frac{\partial v}{\partial p} \, dS \, dt \\
& + \int_{\Omega} \mu_3 \left(\frac{\partial v}{\partial p}(x, 0) - \frac{\partial v_0}{\partial p}(x, 0) \right) dx.
\end{aligned}$$

Po vytknutí výrazů ve tvaru $\frac{\partial u}{\partial p}$ a $\frac{\partial v}{\partial p}$ a přerovnání získáme

$$\begin{aligned}
\frac{dJ^*}{dp} &= \int_{\Omega} \left(\lambda_1 \frac{\partial u}{\partial p}(x, T) + (\lambda_3(x) - \lambda_1(x, 0)) \frac{\partial u}{\partial p}(x, 0) - \lambda_3(x) \frac{\partial u_0}{\partial p}(x, 0) \right) dx \\
&+ \int_0^T \int_{\Omega} \left(u - u_d - \frac{\partial \lambda_1}{\partial t} - D_u \Delta \lambda_1 + \lambda_1 v^2 + \lambda_1 f - \mu_1 v^2 \right) \frac{\partial u}{\partial p} dx dt \\
&+ \int_0^T \int_{\partial \Omega} \lambda_2 \frac{\partial u}{\partial p} dS dt + \int_0^T \int_{\partial \Omega} D_u \left(\frac{\partial \lambda_1}{\partial n} \frac{\partial u}{\partial p} - \lambda_1 \frac{\partial}{\partial p} \frac{\partial u}{\partial n} \right) dS dt \\
&+ \int_0^T \int_{\Omega} \lambda_1 (1 - u, 0) dx dt \\
&+ \int_{\Omega} \left(\mu_1 \frac{\partial v}{\partial p}(x, T) + (\mu_3(x) - \mu_1(x, 0)) \frac{\partial v}{\partial p}(x, 0) - \mu_3(x) \frac{\partial v_0}{\partial p}(x, 0) \right) dx \\
&+ \int_0^T \int_{\Omega} \left(v - v_d - \frac{\partial \mu_1}{\partial t} - D_v \Delta \mu_1 - 2\mu_1 uv + \mu_1 f + \mu_1 k + 2\lambda_1 uv \right) \frac{\partial v}{\partial p} dx dt \\
&+ \int_0^T \int_{\partial \Omega} \mu_2 \frac{\partial v}{\partial p} dS dt + \int_0^T \int_{\partial \Omega} D_v \left(\frac{\partial \mu_1}{\partial n} \frac{\partial v}{\partial p} - \mu_1 \frac{\partial}{\partial p} \frac{\partial v}{\partial n} \right) dS dt \\
&+ \int_0^T \int_{\Omega} \mu_1(v, v) dx dt.
\end{aligned}$$

Výpočtu derivace funkcí u a v podle parametrů p zamezíme splněním následující soustavy rovnic

$$u - u_d - \frac{\partial \lambda_1}{\partial t} - D_u \Delta \lambda_1 + \lambda_1(v^2 + f) - \mu_1 v^2 = 0 \quad \text{na } \Omega \times [0, T], \quad (3.23)$$

$$\lambda_1 = 0 \quad \text{na } \partial \Omega \times [0, T], \quad (3.24)$$

$$\lambda_1(x, T) = 0 \quad \text{na } \Omega, \quad (3.25)$$

$$\lambda_3 - \lambda_1 = 0 \quad \text{na } \Omega, \quad (3.26)$$

$$\lambda_3(x, 0) = 0 \quad \text{na } \Omega, \quad (3.27)$$

$$\lambda_2 = 0 \quad \text{na } \partial \Omega \times [0, T], \quad (3.28)$$

$$v - v_d - \frac{\partial \mu_1}{\partial t} - D_v \Delta \mu_1 - 2\mu_1 uv + \mu_1(f + k) + 2\lambda_1 uv = 0 \quad \text{na } \Omega \times [0, T], \quad (3.29)$$

$$\mu_1 = 0 \quad \text{na } \partial \Omega \times [0, T], \quad (3.30)$$

$$\mu_1(x, T) = 0 \quad \text{na } \Omega, \quad (3.31)$$

$$\mu_3 - \mu_1 = 0 \quad \text{na } \Omega, \quad (3.32)$$

$$\mu_3(x, 0) = 0 \quad \text{na } \Omega, \quad (3.33)$$

$$\mu_2 = 0 \quad \text{na } \partial \Omega \times [0, T]. \quad (3.34)$$

Gradient $\frac{dJ^*}{dp}$ se pak zjednoduší na tvar

$$\frac{dJ^*}{dp} = \int_0^T \int_{\Omega} \left(\lambda_1(1 - u, 0) + \mu_1(v, v) \right) dx dt. \quad (3.35)$$

Poznamenejme však, že vzhledem ke konkrétnímu tvaru gradientu (3.35) se adjungovaná rovnice zredukuje pouze na (3.23)-(3.25) a (3.29)-(3.31).

3.3 Numerické řešení

Podobně jako v předchozí kapitole odvodíme diferenční schéma pro řešení dopředné a adjungované rovnice, gradientu a ztrátové funkce.

- **Diskretizace časového intervalu $[0, T]$:**

Časové hladiny zavedeme zcela analogicky jako v předchozí kapitole

$$\{k\tau \mid \tau > 0, k = 0, \dots, M\},$$

kde $\tau > 0$ je časový krok a $M + 1$ je počet časových hladin.

- **Diskretizace oblasti Ω :**

Pro zjednodušení uvažujme $\Omega = (0, l)^2$, kde $l > 0$ je délka strany ve směru x a y . Pak můžeme definovat síť uzlů $\bar{\omega}_h$

$$\bar{\omega}_h := \{(ih, jh) \mid i, j = 0, \dots, m\},$$

kde $h = \frac{l}{m}$. Dále zaved' me síť vnitřních uzlů ω_h

$$\omega_h := \{(ih, jh) \mid i, j = 1, \dots, m-1\},$$

a hranici sítě $\partial\omega_h$

$$\partial\omega_h := \bar{\omega}_h \setminus \omega_h.$$

Poznamenejme, že přestože jsme doted' pro souřadnici používali jednotnou proměnnou x , v této sekci zavedeme standardní proměnné x a y . Poté pro veličinu tvaru $z = z(x, y, t)$ vystupujících v dopředné nebo adjungované rovnici zavedeme v uzlech sítě zkrácený zápis

$$z_{i,j}^k := z(ih, jh, k\tau),$$

pro $k = 0, \dots, M$, $i, j = 0, \dots, m$.

- **Nahrazení diferenciálních výrazů diferenčními:**

K aproximaci časových derivací použijeme opět dopřednou diferenci. Bud' $z = z(x, y, t)$ veličina, jejíž časovou derivaci chceme aproximovat, náhradu pak navrhneme ve tvaru

$$\frac{dz}{dt}(ih, jh, k\tau) \approx \frac{z_{i,j}^{k+1} - z_{i,j}^k}{\tau}, \quad \text{pro } k = 0, \dots, M-1 \text{ a } i, j = 0, \dots, m. \quad (3.36)$$

Laplaceův operátor $\Delta z = \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2}$ též veličiny nahradíme v obou směrech centrální druhou diferencí

$$\Delta z(ih, jh, k\tau) \approx \frac{z_{i-1,j}^k + z_{i+1,j}^k + z_{i,j-1}^k + z_{i,j+1}^k - 4z_{i,j}^k}{h^2}, \quad \text{pro } k = 0, \dots, M-1 \text{ a } i, j = 1, \dots, m-1.$$

Nyní můžeme přikročit k sestavení diferenčních schémat.

3.3.1 Dopředná rovnice

Připomeňme, že dopředná úloha má tvar

$$\frac{\partial u}{\partial t} = D_u \Delta u - uv^2 + f(1 - u) \quad \text{na } \Omega \times [0, T], \quad (3.37)$$

$$\frac{\partial v}{\partial t} = D_v \Delta v + uv^2 - (f + k)v \quad \text{na } \Omega \times [0, T], \quad (3.38)$$

$$u = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.39)$$

$$v = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.40)$$

$$u(x, 0) = u_0(x) \quad \text{na } \Omega, \quad (3.41)$$

$$v(x, 0) = v_0(x) \quad \text{na } \Omega. \quad (3.42)$$

Dosažením diferenčních náhrad získáme pro $k = 0, \dots, M - 1$ na ω_h soustavu algebraických rovnic tvaru

$$u_{i,j}^{k+1} = u_{i,j}^k + \tau \left(D_u \frac{u_{i-1,j}^k + u_{i+1,j}^k + u_{i,j-1}^k + u_{i,j+1}^k - 4u_{i,j}^k}{h^2} - u_{i,j}^k (v_{i,j}^k)^2 + f(1 - u_{i,j}^k) \right), \quad (3.43)$$

$$v_{i,j}^{k+1} = v_{i,j}^k + \tau \left(D_v \frac{v_{i-1,j}^k + v_{i+1,j}^k + v_{i,j-1}^k + v_{i,j+1}^k - 4v_{i,j}^k}{h^2} + u_{i,j}^k (v_{i,j}^k)^2 - (f + k)v_{i,j}^k \right). \quad (3.44)$$

Tento tvar opět odpovídá Eulerově metodě, podobně jako v předchozí kapitole však k řešení použijeme Adamsovu-Bashforthovu metodu druhého řádu. Přepíšeme-li rovnice (3.43) a (3.44) do tvaru

$$u_{i,j}^{k+1} = u_{i,k}^k + \tau f_{i,j}^k, \quad (3.45)$$

$$v_{i,j}^{k+1} = v_{i,k}^k + \tau g_{i,j}^k, \quad (3.46)$$

pak Adamsova-Bashforthova metoda bude mít pro $k = 1, \dots, M - 1$ na ω_h předpis

$$u_{i,j}^{k+1} = u_{i,k}^k + \frac{\tau}{2} \left(3f_{i,j}^k - f_{i,j}^{k-1} \right), \quad (3.47)$$

$$v_{i,j}^{k+1} = v_{i,k}^k + \frac{\tau}{2} \left(3g_{i,j}^k - g_{i,j}^{k-1} \right), \quad (3.48)$$

s okrajovými podmínkami pro $k = 0, \dots, M$ na $\partial\omega_h$

$$u_{i,j}^k = 0, \quad (3.49)$$

$$v_{i,j}^k = 0, \quad (3.50)$$

a počátečními podmínkami na $\overline{\omega_h}$

$$u_{i,j}^0 = u_{0,i,j}, \quad (3.51)$$

$$v_{i,j}^0 = v_{0,i,j}, \quad (3.52)$$

$$u_{i,j}^1 = u_{i,j}^0 + \tau f_{i,j}^0, \quad (3.53)$$

$$v_{i,j}^1 = v_{i,j}^0 + \tau g_{i,j}^0. \quad (3.54)$$

3.3.2 Adjungovaná rovnice

Adjungovaná úloha pro μ_1 a λ_1 má tvar

$$u - u_d - \frac{\partial \lambda_1}{\partial t} - D_u \Delta \lambda_1 + \lambda_1(v^2 + f) - \mu_1 v^2 = 0 \quad \text{na } \Omega \times [0, T], \quad (3.55)$$

$$v - v_d - \frac{\partial \mu_1}{\partial t} - D_v \Delta \mu_1 - 2\mu_1 uv + \mu_1(f + k) + 2\lambda_1 uv = 0 \quad \text{na } \Omega \times [0, T], \quad (3.56)$$

$$\lambda_1 = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.57)$$

$$\mu_1 = 0 \quad \text{na } \partial\Omega \times [0, T], \quad (3.58)$$

$$\lambda_1(x, T) = 0 \quad \text{na } \Omega, \quad (3.59)$$

$$\mu_1(x, T) = 0 \quad \text{na } \Omega. \quad (3.60)$$

Náhradou diferenciálních výrazů dostaneme diferenční schéma pro adjungovanou rovnici na ω_h tvaru

$$\lambda_{1,i,j}^{M-k-1} = \lambda_{1,i,j}^{M-k} + \tau \left(u_{i,j}^{M-k} - u_{d,i,j}^{M-k} - D_u \frac{\lambda_{1,i-1,j}^{M-k} + \lambda_{1,i+1,j}^{M-k} + \lambda_{1,i,j-1}^{M-k} + \lambda_{1,i,j+1}^{M-k} - 4\lambda_{1,i,j}^{M-k}}{h^2} + \right. \quad (3.61)$$

$$\left. \lambda_{1,i,j}^{M-k}((v_{i,j}^{M-k})^2 + f) - \mu_{1,i,j}^{M-k}(v_{i,j}^{M-k})^2 \right), \quad (3.62)$$

$$\mu_{1,i,j}^{M-k-1} = \mu_{1,i,j}^{M-k} + \tau \left(v_{i,j}^{M-k} - v_{d,i,j}^{M-k} - D_v \frac{\mu_{1,i-1,j}^{M-k} + \mu_{1,i+1,j}^{M-k} + \mu_{1,i,j-1}^{M-k} + \mu_{1,i,j+1}^{M-k} - 4\mu_{1,i,j}^{M-k}}{h^2} + \right. \quad (3.63)$$

$$\left. \mu_{1,i,j}^{M-k}(f + k) + 2\lambda_{1,i,j}^{M-k} u_{i,j}^{M-k} v_{i,j}^{M-k} \right). \quad (3.64)$$

Podobně jako u dopředné rovnice k řešení použijeme Adamsovu-Bashforthovu metodu. Přepíšeme-li opět předchozí schéma do tvaru

$$\lambda_{1,i,j}^{M-k-1} = \lambda_{1,i,j}^{M-k} + \tau \hat{f}_{i,j}^{M-k}, \quad (3.65)$$

$$\mu_{1,i,j}^{M-k-1} = \mu_{1,i,j}^{M-k} + \tau \hat{g}_{i,j}^{M-k}, \quad (3.66)$$

můžeme diferenční schéma zapsat pro $k = 1, \dots, M-1$ na $\overline{\omega_h}$ ve tvaru

$$\lambda_{1,i,j}^{M-k-1} = \lambda_{1,i,j}^{M-k} + \frac{\tau}{2} \left(3\hat{f}_{i,j}^{M-k} - \hat{f}_{i,j}^{M-k+1} \right), \quad (3.67)$$

$$\mu_{1,i,j}^{M-k-1} = \mu_{1,i,j}^{M-k} + \frac{\tau}{2} \left(3\hat{g}_{i,j}^{M-k} - \hat{g}_{i,j}^{M-k+1} \right), \quad (3.68)$$

s okrajovými podmínkami pro $k = 0, \dots, M$ na $\partial\omega_h$

$$\lambda_1^k = 0, \quad (3.69)$$

$$\mu_1^k = 0, \quad (3.70)$$

a podmínkami v posledních dvou časových hladinách na $\overline{\omega_h}$

$$\lambda_1^{M-1} = 0, \quad (3.71)$$

$$\mu_1^{M-1} = 0, \quad (3.72)$$

$$\lambda_1^{M-2} = \tau \hat{f}^{M-1}, \quad (3.73)$$

$$\mu_1^{M-2} = \tau \hat{g}^{M-1}. \quad (3.74)$$

3.3.3 Gradient

Nakonec musíme odvodit aproximaci pro výpočet gradientu

$$\frac{dJ^*}{dp} = \int_0^T \int_{\Omega} \lambda_1(1-u, 0) + \mu_1(v, v) \, dx \, dt. \quad (3.75)$$

K tomu použijeme analogii lichoběžníkové metody ve 2D. Pro veličinu z definovanou na síti $\bar{\omega}_h$ definujeme průměry čtyř sousedních uzlů jako

$$z_{i,j}^{k,\text{avg}} := \frac{1}{4}(z_{i,j}^k + z_{i+1,j}^k + z_{i,j+1}^k + z_{i+1,j+1}^k), \quad (3.76)$$

pro $k = 0, \dots, M$ a $i, j = 0, \dots, m-1$. S tímto značením pak můžeme integrál aproximovat jako

$$\int_0^T \int_{\Omega} \lambda_1(1-u, 0) + \mu_1(v, v) \, dx \, dt \approx \sum_{k=0}^M \sum_{i,j=0}^{m-1} \left(\lambda_{1,i,j}^{k,\text{avg}} (1 - u_{i,j}^{k,\text{avg}}) + \mu_{1,i,j}^{k,\text{avg}} v_{i,j}^{k,\text{avg}}, \mu_{1,i,j}^{k,\text{avg}} v_{i,j}^{k,\text{avg}} \right) h^2. \quad (3.77)$$

3.3.4 Ztrátová funkce

Podobně jako v předchozí sekci odvodíme předpis pro aproximaci ztrátové funkce (3.7). Integrál můžeme aproximovat předpisem

$$\frac{1}{2} \int_0^T \int_{\Omega} \left((u - u_d)^2 + (v - v_d)^2 \right) \, dx \, dt \approx \sum_{k=0}^M \sum_{i,j=0}^{m-1} \left(\left(u_{i,j}^{k,\text{avg}} - u_{d,i,j}^{k,\text{avg}} \right)^2 + \left(v_{i,j}^{k,\text{avg}} - v_{d,i,j}^{k,\text{avg}} \right)^2 \right) h^2. \quad (3.78)$$

3.4 Implementace

Pro úplnost dodejme analogii algoritmu 2 pro tuto část. Algoritmus používaný k výpočtu gradientu pomocí konečných diferencí pak bude zcela analogický algoritmu 3.

Algoritmus 4: Výpočet gradientu adjungovanou rovnicí.

Vstup: Data $d(t)$

Vstup: Parametry α, ρ algoritmu RMSprop

Vstup: Počáteční parametry p_0

Vstup: Prahová hodnota

- 1: **while** Hodnota ztrátové funkce > prahová hodnota **do**
 - 2: Vyřeš dopředné rovnice (3.3), (3.4) pro u a v
 - 3: Vyřeš adjungované rovnice (3.23), (3.29) pro λ_1 a μ_1
 - 4: Vypočti gradient (3.35) a normalizuj ho
 - 5: Proveď krok v RMSprop algoritmu (1.7)
 - 6: Vypočti hodnotu ztrátové funkce podle (3.7)
 - 7: **end while**
-

3.5 Výpočetní studie

V této sekci ukážeme výpočty dopředné rovnice s různými kombinacemi parametrů f a k , abychom ilustrovali různé druhy vzorů, které lze získat pomocí Grayova-Scottova modelu. Protože diskretizaci jednotlivých diferenciálních operátorů jsme si již připravili, můžeme přistoupit k samotné simulaci.

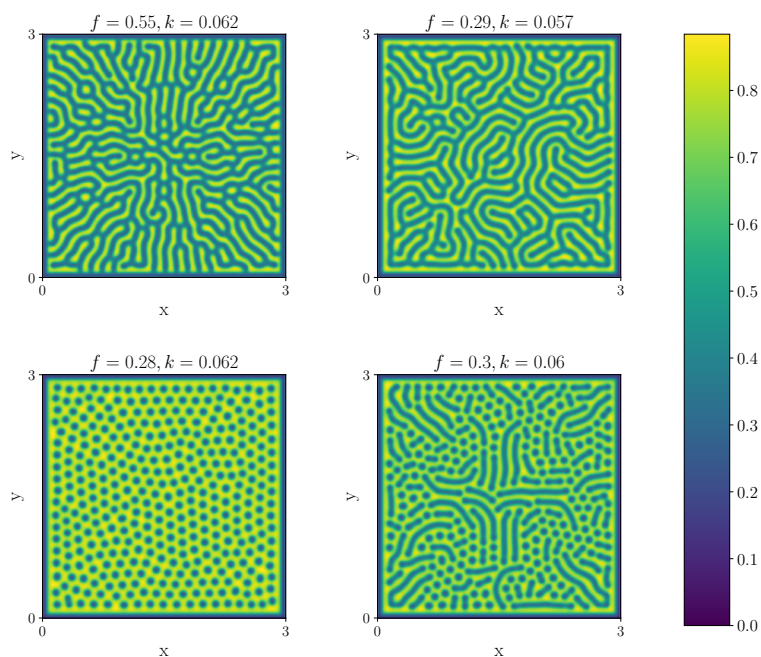
Poznamenejme, že nastavení simulace je inspirováno článkem [6], které zde v krátkosti shrneme. Pro difúzní koeficienty použijeme hodnoty $D_u = 2 \times 10^{-5}$ a $D_v = 1 \times 10^{-5}$. Na počátku simulace nastavíme hodnoty funkcí jako $u = 1$ a $v = 0$ na $\overline{\omega}_h$. Poté nastavíme hodnoty ve čtvercové oblasti kolem středu $\overline{\omega}_h$, jako $u = 0.5$ a $v = 0.25$, čímž simulujeme počátek chemické reakce. Dále tuto oblast kolem středu perturbujeme přičtením náhodně vygenerovaného čísla mezi $(-0.05, 0.05)$, abychom narušili symetrii. Jako numerickou metodu volíme Adamsovu-Bashforthovu metodu.

Použité parametry simulace jsou shrnuty v tabulce 3.1.

Tabulka 3.1: Parametry simulace.

Parametr	Význam	Hodnota
l	Délka strany	3.0
m	Počet uzlů v ose x a y	301
h	Prostorový krok	1/300
T	Konečný čas	10000
τ	Časový krok	0.25

Výsledky simulací pro různé kombinace parametrů jsou na obrázku 3.1. Obrázky zachycují výslednou koncentraci sloučeniny u .



Obrázek 3.1: Grayův-Scottův model pro různé parametry.

Nyní můžeme přejít k odhadu parametrů pomocí adjungované rovnice a konečných diferencí. Odhad parametrů povedeme podobně jako v předchozí kapitole s tím rozdílem, že nyní vyzkoušíme více kombinací parametrů dat f a k , konkrétně použijeme dvojice, které jsme již využili u simulace dopředné rovnice.

Vzhledem ke znalosti typických hodnot parametrů f a k , které se pohybují v relativně malé oblasti budeme moci u tohoto modelu získat poměrně dobré počáteční odhady p_0 , konkrétně budeme volit počáteční hodnoty parametrů $f, k \in [0.03, 0.05]$.

1. Výpočet gradientu pomocí adjungované rovnice

Budeme postupovat podle algoritmu 4. Podobně jako v předchozí kapitole opět použijeme výrazně kratší časový interval. Jelikož si také z podstaty adjungované rovnice musíme do paměti ukládat hodnoty dopředné rovnice ve všech bodech $\overline{\omega}_h$ a časových hladinách, budeme muset zmenšit i rozlišení schématu. Tento problém by se samozřejmě dal vyřešit spuštěním programu na počítači s větší pamětí, je však dobré mít tento fakt na paměti, protože u parciálních diferenciálních rovnic musíme, na rozdíl od těch obyčejných, ukládat mnohem více hodnot a s rostoucím rozlišením a délkou intervalu pak množství dat k uložení narůstá velmi rychle.

Časový interval $[0, T]$ musíme opět zvolit takovým způsobem, aby konečný čas T nebyl příliš malý, a tedy abychom byli schopni získat z dat dostatek informací a současně, aby nebyl příliš velký, a tudíž by adjungovaná rovnice začala divergovat. Opět se nezdá, že by existoval jednoduchý způsob, jak vhodný konečný čas T zvolit a je nutné vyzkoušet několik možností a nejprve získat zkušenost s chováním adjungované rovnice.

Použité parametry simulace jsou shrnuty v tabulce 3.2

Tabulka 3.2: Parametry simulace.

Parametr	Význam	Hodnota
l	Délka strany	1.0
m	Počet uzlů v ose x a y	51
h	Prostorový krok	1/50
T	Konečný čas	40
τ	Časový krok	0.1

Jako numerickou metodu použijeme Adamsovu-Bashforthovu metodu. Dále upravíme parametry algoritmu RMSprop, kdy zvolíme $\alpha = 10^{-5}$ a $\rho = 0.99995$. K normalizaci hodnot gradientu použijeme nejprve logaritmickou a poté klasickou normalizaci.

Výsledky jsou shrnuty v tabulkách 3.3 - 3.6.

Tabulka 3.3: Výsledek - $p_{\text{data}} = (0.055, 0.062)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	2.9×10^{-10}	(0.0550008, 0.0619997)
(0.04, 0.04)	9.5×10^{-10}	(0.0550014, 0.0619995)
(0.05, 0.05)	9.9×10^{-10}	(0.0550012, 0.0619992)
(0.03, 0.05)	8.6×10^{-10}	(0.0550015, 0.0619993)

Tabulka 3.4: Výsledek - $p_{\text{data}} = (0.029, 0.057)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	2.3×10^{-10}	(0.0290451, 0.0570002)
(0.04, 0.04)	9.6×10^{-10}	(0.0289988, 0.0570002)
(0.05, 0.05)	7.4×10^{-10}	(0.0289997, 0.0569994)
(0.03, 0.05)	1.5×10^{-10}	(0.0290003, 0.0569997)

Tabulka 3.5: Výsledek - $p_{\text{data}} = (0.028, 0.062)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	8.1×10^{-11}	(0.028, 0.0619998)
(0.04, 0.04)	6.0×10^{-10}	(0.0280007, 0.0619999)
(0.05, 0.05)	1.3×10^{-10}	(0.0279998, 0.0619998)
(0.03, 0.05)	3.0×10^{-8}	(0.0279949, 0.0620045)

Tabulka 3.6: Výsledek - $p_{\text{data}} = (0.03, 0.06)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	7.2×10^{-10}	(0.0300007, 0.0599993)
(0.04, 0.04)	9.4×10^{-10}	(0.0300004, 0.0600006)
(0.05, 0.05)	4.8×10^{-10}	(0.0299995, 0.0599997)
(0.03, 0.05)	1.5×10^{-7}	(0.0300126, 0.0599904)

Jak vidíme z tabulek, po příslušném nastavení parametrů simulace a algoritmu RMSprop se nám podařilo parametry odhadnout, nicméně nalezení správného nastavení všech parametrů není triviální.

CPU časy výpočtů se pohybovaly kolem dvou minut.

2. Výpočet gradientu metodou konečných diferencí

Nyní budeme opět postupovat podle algoritmu 3. Jelikož přístup přes konečné diference nemá problémy s adjungovanou rovnicí, můžeme zvolit parametry simulace volněji. Pro možnost porovnání však nastavení simulace ponechme stejné jako u odhadu pomocí adjungované rovnice. Jedinou změnu uděláme v tom, že od začátku budeme používat klasickou normalizaci gradientu.

Výsledky jsou shrnuty v tabulkách 3.7 - 3.10.

Tabulka 3.7: Výsledek - $p_{\text{data}} = (0.055, 0.062)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	7.3×10^{-10}	(0.0550002, 0.0620002)
(0.04, 0.04)	9.9×10^{-10}	(0.0549998, 0.0619998)
(0.05, 0.05)	6.6×10^{-10}	(0.0549999, 0.0620003)
(0.03, 0.05)	1.7×10^{-10}	(0.0550005, 0.0619997)

Tabulka 3.8: Výsledek - $p_{\text{data}} = (0.029, 0.057)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	2.3×10^{-9}	(0.0290008, 0.0570008)
(0.04, 0.04)	2.3×10^{-9}	(0.0290008, 0.0570008)
(0.05, 0.05)	2.5×10^{-9}	(0.0289991, 0.0569991)
(0.03, 0.05)	9.9×10^{-10}	(0.0289994, 0.0569994)

Tabulka 3.9: Výsledek - $p_{\text{data}} = (0.028, 0.062)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	9.9×10^{-10}	(0.0279995, 0.0619994)
(0.04, 0.04)	9.2×10^{-10}	(0.028, 0.0619993)
(0.05, 0.05)	9.8×10^{-10}	(0.0280004, 0.0620006)
(0.03, 0.05)	5.2×10^{-9}	(0.0280012, 0.0620012)

Tabulka 3.10: Výsledek - $p_{\text{data}} = (0.03, 0.06)$.

Počáteční parametry	Hodnota ztrátové funkce	Konečné parametry
(0.03, 0.03)	9.9×10^{-10}	(0.0300005, 0.0600005)
(0.04, 0.04)	9.4×10^{-9}	(0.0299984, 0.0599984)
(0.05, 0.05)	9.9×10^{-10}	(0.0299995, 0.0599995)
(0.03, 0.05)	1.5×10^{-8}	(0.0300126, 0.0599904)

CPU časy výpočtů se v tomto případě pohybovaly kolem minuty.

3.6 Diskuze výsledků

Jak jsme mohli vidět v předchozí sekci, po správném nastavení simulací se podařilo parametry odhadnout oběma způsoby. Oproti předchozí kapitole, kdy se odhad pomocí konečných diferencí zdál mnohem výhodnější, se nyní ukázal odhad pomocí adjungované rovnice srovnatelně dobrý s konečnými diferencemi. Jak už jsme však zmínili, je nutné dodat, že přístup přes adjungovanou rovnici je citlivější na správné nastavení. Jelikož tento model opět obsahuje malý počet parametrů (2), adjungovaná rovnice pro tuto úlohu plně nevyužije svůj potenciál, který teoreticky nabízí a nevyvažuje složitější nastavení oproti přístupu přes konečné diference.

Závěr

V této práci jsme zkoumali možnost odhadu parametrů matematických modelů popsaných pomocí soustav obyčejných a parciálních diferenciálních rovnic z dat, konkrétně Volterrova-Lotkova modelu, SIR modelu, Lorenzova modelu a Grayova-Scottova modelu. K tomu jsme představili optimalizační úlohu, kterou jsme řešili pomocí gradientního algoritmu RMSprop. K výpočtu gradientu jsme použili výpočet pomocí adjungované rovnice a přímý výpočet metodou konečných diferencí. Výsledné algoritmy jsme implementovali v programovacím jazyce C++.

V první kapitole jsme představili krátký přehled metod používaných v optimalizaci a strojovém učení.

Ve druhé kapitole jsme nejprve odvodili adjungovanou rovnici pro obecný model popsaný pomocí obyčejných diferenciálních rovnic. Dále jsme ukázali, jak vypočítat gradient pomocí metody konečných diferencí. Poté jsme krátce představili jednotlivé modely, jimiž jsme se zabývali a odvodili příslušné adjungované rovnice. Dále jsme odvodili diferenční schéma pro obecný model a naznačili způsob numerického řešení rovnic, k čemuž jsme využili Adamsovu-Bashforthovu metodu druhého řádu. Následně jsme ve výpočetní sekci provedli výpočty pro všechny modely a poté jsme provedli samotný odhad parametrů pomocí obou metod. Ukázali jsme, že pomocí adjungované rovnice lze po příslušném nastavení odhadnout parametry pro Volterrův-Lotkův a SIR model. Pro Lorenzův model se odhad pomocí adjungované rovnice nepovedl. Pomocí metody konečných diferencí jsme odhadli parametry pro všechny tři modely.

Ve třetí kapitole jsme představili Grayův-Scottův model reakčně difúzních rovnic a poté jsme pro něj odvodili adjungovanou rovnici. Následně jsme odvodili diferenční schéma pro numerické řešení rovnic, ke kterému jsme opět použili Adamsovu-Bashforthovu metodu druhého řádu. Poté jsme řešili Grayův-Scottův model pro různé kombinace parametrů, které jsme se dále snažili odhadnout. Po správném nastavení se ukázalo, že parametry lze odhadnout pomocí adjungované rovnice i konečných diferencí.

Na tuto práci by šlo dále navázat zejména zkoumáním toho, jak správně nastavovat numerické výpočty, parametry gradientních metod a ostatní nastavení výpočtů, abychom dosáhli jednak robustnějších výsledků vzhledem k počátečnímu odhadu parametrů, a jednak optimalizací trvání samotného výpočtu.

Literatura

- [1] Charu C Aggarwal, Lagerstrom-Fife Aggarwal, and Lagerstrom-Fife. *Linear algebra and optimization for machine learning*, volume 156. Springer, 2020.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [3] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.
- [4] James D. Murray. *Mathematical Biology I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2002.
- [5] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.
- [6] John E Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.
- [7] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.
- [8] Biswa Sengupta, Karl J Friston, and William D Penny. Efficient gradient computation for dynamical models. *NeuroImage*, 98:521–527, 2014.
- [9] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [10] Walter A Strauss. *Partial differential equations: An introduction*. John Wiley & Sons, 2007.