

# Problem optimizacije slabo određenih modela nuklearne strukture

---

Imbrišak, Marko

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:657481>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-24**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

Marko Imbrišak

PROBLEM OPTIMIZACIJE SLABO ODREĐENIH  
MODELA NUKLEARNE STRUKTURE

Diplomski rad

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ  
FIZIKA; SMJER ISTRAŽIVAČKI

**Marko Imbrišak**

Diplomski rad

**Problem optimizacije slabo određenih  
modela nuklearne strukture**

Voditelj diplomskog rada: prof. dr. sc. Tamara Nikšić

Ocjena diplomskog rada: \_\_\_\_\_

Povjerenstvo: 1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Datum polaganja: \_\_\_\_\_

Zagreb, 2019.

Ovim bih se putem želio zahvaliti prof. dr. sc. Tamari Nikšić na njezinim prijedlozima, komentarima i strpljenju. Također, zahvalio bih se dipl. ing. mat. Zrinki Vidović-Tisanić na njezinim savjetima i zapažanjima. I na kraju svojim prijateljima Krešimiru i Ratku što su vrijeme studiranja činili ugodnijim.

## Sažetak

Modeli koji se koriste za opis većine prirodnih pojava imaju velik broj slobodnih parametara čije vrijednosti određujemo prilagodbom teorijskih predviđanja modela empirijskim podacima. U jednostavnim modelima, promjena bilo kojeg parametra znatno utječe na kvalitetu prilagodbe predviđanja empirijskim podacima, no također postoje modeli koji sadrže i aljkavo odredive parametre. To su određeni parametri i/ili kombinacije parametara koje se mogu značajno promijeniti uz minimalnu promjenu kvalitete prilagodbe. Nedavno je razvijena aproksimacijska metoda mnogostrukosti s rubom (MBAM) kojome je moguće eliminirati što veći broj aljkavih stupnjeva slobode modela, uz minimalan gubitak na preciznosti same prilagodbe. Problem MBAM metode jest u tome što zahtijeva izračun numeričkih derivacija modela. U ovome smo diplomskom radu razvili računalnu implementaciju MBAM metode poboljšane tehnikama algoritamske diferencijacije kako bi uklonili numeričke greške nastale zbog numeričke diferencijacije. Računalnu implementaciju smo testirali na širokom rasponu modela zadanih ili kao analitički izrazi ili rješenja sustava diferencijalnih jednačbi. U konačnici provjerili smo postoji li razlika prilikom korištenja preciznijih derivacija u MBAM redukciji na modelu atomske jezgre zasnovanom na nuklearnom energijskom funkcionalu gustoće DD-PC1.

Ključne riječi: aproksimacijska metoda mnogostrukosti s rubom, atomske jezgre, energijski funkcionali gustoće

# Optimization problem of sloppy nuclear structure models

## Abstract

Models used in describing most physical phenomena have a large number of free parameters, usually estimated by fitting theoretical predictions to empirical data. In simple models, changes in any model parameter would produce a visible change in the quality of fit. More complex models also have degrees of freedom, called sloppy parameters, that can be significantly altered without disrupting the quality of the fit. A novel method, called the manifold boundary approximation method (MBAM), has been shown to be able to remove sloppy parameters without affecting the quality of model predictions. However, the MBAM method requires a large number of numerical derivatives, raising the question of numerical errors. In this diploma thesis, we have developed codes that improve the MBAM method using algorithmic differentiation instead of numerical approximations. We have tested this implementation on a wide variety of models, from simple analytical expressions to complex systems of differential equations. Finally, we have tested whether using algorithmic differentiation impacts results of the MBAM method applied to the model of the atomic nucleus based on the nuclear energy density functional DD-PC1.

Keywords: The Manifold Boundary Approximation Method, Atomic Nuclei, Energy Density Functionals

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Diferencijalna geometrija</b>	<b>2</b>
2.1	Pregled pojmova iz topologije . . . . .	2
2.2	Definicija mnogostrukosti . . . . .	3
2.3	Tangentni prostor . . . . .	4
2.4	Metrika . . . . .	10
2.5	Kovarijantna derivacija . . . . .	12
2.6	Riemannov tenzor . . . . .	14
2.7	Riccijev tenzor i skalar . . . . .	14
<b>3</b>	<b>Statistika</b>	<b>15</b>
3.1	Prostor mjere . . . . .	15
3.2	Prostor vjerojatnosti . . . . .	19
3.3	Fisherova informacijska metrika . . . . .	24
<b>4</b>	<b>Prostor mjerenja i prostor parametara</b>	<b>27</b>
4.1	Optimalno računanje Christoffelovih simbola . . . . .	29
4.2	Aproksimacijska metoda mnogostrukosti s rubom . . . . .	30
4.3	Zakrivljenost . . . . .	30
4.3.1	Intrinzična zakrivljenost . . . . .	30
4.3.2	Ekstrinzična zakrivljenost . . . . .	32
<b>5</b>	<b>Računalna implementacija</b>	<b>34</b>
5.1	Algoritamska diferencijacija . . . . .	34
5.2	Analitički rješivi primjeri . . . . .	35
5.3	Modeli konačne zakrivljenosti . . . . .	42
5.4	Primjer MBAM redukcije na primjeru razlomljenog modela . . . . .	45
5.5	Primjeri kompleksnih dinamičkih sistema . . . . .	48
5.5.1	Troparametarsko njihalo . . . . .	48
5.5.2	FitzHugh-Nagumo model . . . . .	55
5.5.3	Lorenzov model . . . . .	57

<b>6</b>	<b>Primjena MBAM metode nuklearne modele</b>	<b>60</b>
6.1	Komponente Diracove jednačbe . . . . .	61
6.2	Primjena automatske diferencijacije na funkcional DD-PC1 . . . . .	62
<b>7</b>	<b>Zaključak</b>	<b>68</b>
	<b>Literatura</b>	<b>69</b>
<b>A</b>	<b>Dodatak: Python implementacija MBAM metode</b>	<b>71</b>
A.1	Kod za MBAM metodu . . . . .	71
A.2	Kod za MBAM metodu na funkcionalu DDPC-1 . . . . .	90



# 1 Uvod

Modeli koji se koriste za opis većine prirodnih pojava imaju velik broj slobodnih parametara. Parametre modela možemo odrediti postupkom prilagodbe teorijskih predviđanja modela empirijskim podacima, najčešće u obliku minimizacije  $\chi^2$  vrijednosti. Nažalost, često nemamo na raspolaganju dovoljno podataka za precizno određivanje svih stupnjeva slobode modela, što dovodi do pojave da su neke kombinacije parametara modela slabo određene. Nadalje, čak i ako imamo velik skup mjerenja, sami modeli mogu biti takvi da je  $\chi^2$  vrijednost osjetljiva samo na promjene određenih parametara, dok promjene drugih parametara ne utječu znatno na  $\chi^2$  vrijednost. Modele koji u procesu pronalaženja najbolje prilagodbe pokazuju eksponencijalni raspon osjetljivosti na promjene parametara zovemo aljkavi modeli<sup>1</sup>. U posljednje vrijeme pokrenuta su sustavna istraživanja mogućnosti eliminiranja slabo određenih stupnjeva slobode modela. Glavna je motivacija ovakvih istraživanja da se eliminira što veći broj aljkavih stupnjeva slobode, uz zadržavanje kvalitetnog opisa promatrane pojave. Ovakav postupak ima izrazito velik potencijal za primjene u raznim područjima, od prirodnih preko biomedicinskih pa sve do društvenih znanosti [1–4].

Ovaj diplomski rad predstavlja nastavak razvoja mogućnosti primjene koncepata informacijske geometrije u analizi semi-empirijskih nuklearnih energijskih funkcionala gustoće u vidu aproksimacijske metode mnogostrukosti s rubom. Razvili smo računalnu implementaciju aproksimacijske metode mnogostrukosti s rubom koja zaobilazi ograničenja i nepreciznosti povezane s numeričkim aproksimacijama pri evaluaciji derivacija koristeći metode algoritamske diferencijacije. Testirali smo mogućnosti razvijenog koda na velikom broju kompliciranih modela, od jednostavnih analitičkih prilagodbi do kaotičnih sustava te sustava s velikim intrinzičnim i ekstrinzičnim zakrivljenostima prostora parametara modela [5]. Posebno smo promotrili nuklearni energijski funkcional gustoće DD-PC1 te smo povezali numeričke probleme pri redukciji modela temeljenom na funkcionalu DD-PC1 s geometrijskim svojstvima modela.

---

<sup>1</sup>eng. sloppy models

## 2 Diferencijalna geometrija

U ovome poglavlju dajemo pregled osnovnih topoloških pojmova te dajemo osnovne izvode vezane za diferencijalnu geometriju, temeljen na [6–8].

### 2.1 Pregled pojmova iz topologije

**Definicija 2.1** (Topologija). Za skup  $X$ , topologija na skupu  $X$ ,  $\mathcal{T}$ , je skup svih podskupova skupa  $X$  sa svojstvima

1.  $X \in \mathcal{T}$  i  $\emptyset \in \mathcal{T}$ .
2. Za konačno mnogo podskupova,  $U_1, \dots, U_n \in \mathcal{T}$ , vrijedi  $\bigcap_{i=1}^n U_i \in \mathcal{T}$ .
3. Za konačno ili beskonačno mnogo podskupova,  $U_1, \dots \in \mathcal{T}$  vrijedi  $\bigcup_i U_i \in \mathcal{T}$ .

Skup  $U \in \mathcal{T}$  naziva se otvoreni skup.

**Definicija 2.2** (Topološki prostor). Par skupa  $X$  i njegove topologije  $\mathcal{T}$ ,  $(X, \mathcal{T})$  zove se topološki prostor.

**Definicija 2.3** (Okolina točke). Okolina točke  $p \in X$  je otvoreni skup  $U$  takav da vrijedi  $p \in U$ .

**Definicija 2.4** (Limes). Neka je  $(x_i)_{i=1}^{\infty}$  niz točaka u topološkom prostoru  $X$ , i  $x \in X$ . Niz  $(x_i)_{i=1}^{\infty}$  konvergira u  $x$  ako za svaku okolinu  $U$  točke  $x$  postoji  $N \in \mathbb{N}$  takav da je  $x_i \in U$  za sve  $i \geq N$ .

**Definicija 2.5** (Neprekidno preslikavanje). Ako su  $X$  i  $Y$  topološki prostori, preslikavanje  $f : X \rightarrow Y$  je neprekidno ako za svaki otvoreni skup  $U \subseteq Y$  vrijedi da je  $f^{-1}(U)$  otvoren u  $X$ .

**Definicija 2.6** (Homeomorfizam). Ako su  $X$  i  $Y$  topološki prostori, homeomorfizam je bijektivno preslikavanje  $\varphi : X \rightarrow Y$  tako da su  $\varphi$  i  $\varphi^{-1}$  neprekidni.

**Definicija 2.7** (Hausdorffov prostor). Topološki prostor  $(X, \mathcal{T})$  zove se Hausdorffov prostor ako za svake dvije točke  $p_1, p_2 \in X$  postoje okoline  $U_1$  i  $U_2$ , respektivno, sa svojstvom  $U_1 \cap U_2 = \emptyset$ .

**Definicija 2.8** (Baza topologije). Skup podskupova od  $X$ ,  $\mathcal{B}$ , je baza topologije od  $X$  ako vrijedi

1. Svaki element iz  $\mathcal{B}$  ujedno je otvoreni skup u  $X$ .
2. Svaki otvoreni skup u  $X$  ujedno je unija elemenata skupa  $\mathcal{B}$ .

Nadalje, za topološki prostor  $(X, \mathcal{T})$  i točku  $p \in X$ , skup okolina točke  $p$ ,  $\mathcal{B}_p$ , zove se baza okoline točke  $p$ .

**Definicija 2.9** (Prva prebrojivost). Topološki prostor  $(X, \mathcal{T})$  je prvo prebrojiv ako za svaki  $p \in X$  postoji prebrojiva baza okoline,  $\mathcal{B}_p$ .

**Definicija 2.10** (Druga prebrojivost). Topološki prostor  $(X, \mathcal{T})$  je drugo prebrojiv ako postoji prebrojiva baza njegove topologije,  $\mathcal{T}$ .

## 2.2 Definicija mnogostrukosti

**Definicija 2.11** (Lokalno euklidski prostor). Topološki prostor  $M$  je lokalno euklidski dimenzije  $n$  ako svaka točka iz  $M$  ima okolinu koja je homeomorfna otvorenom podskupu od  $\mathbb{R}^n$ .

**Definicija 2.12** (Mnogostrukost). Topološka mnogostrukost dimenzije  $n$  je drugo prebrojiv Hausdorffov prostor koji je lokalno euklidski prostor dimenzije  $n$ .

**Definicija 2.13** (Mnogostrukost s rubom). Mnogostrukost s rubom dimenzije  $n$  je drugo prebrojiv Hausdorffov prostor čija svaka točka ima okolinu koja je ili homeomorfna otvorenom skupu u  $\mathbb{R}^n$  ili zatvorenom poluprostoru  $\mathbb{H}^n$ , gdje je

$$\mathbb{H}^n = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_i \geq 0\}.$$

Iako sama mnogostrukost s rubom nije mnogostrukost, njezin unutrašnji dio,  $\text{int}M$ , je otvoren skup te je stoga  $\text{int}M$   $n$ -dimenzionalna mnogostrukost.

Drugi pojam koji se veže uz mnogostrukosti je glatkost te ona omogućuje uvođenje lokalnih koordinata na mnogostrukost koristeći pojam koordinatne karte.

**Definicija 2.14** (Glatko preslikavanje). Preslikavanje  $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$ ,  $F = (f^1, \dots, f^k)$ , je glatko ako su mu sve komponente,  $f^i : \mathbb{R}^n \rightarrow \mathbb{R}$ , klase  $C^\infty$ .

**Definicija 2.15** (Koordinatna karta). Ako je  $M$  lokalno euklidski prostor dimenzije  $n$ , podskup  $U \subseteq M$  naziva se koordinatna domena ako je homeomorfan otvorenom skupu

u  $\mathbb{R}^n$ . Tada se svaki homeomorfizam,  $\varphi$ , sa  $U$  na otvoreni podskup od  $\mathbb{R}^n$  zove koordinatna mapa, a par  $(U, \varphi)$  koordinatna karta od  $M$ . Skup  $U$  je tada koordinatna/euklidska okolina točke  $p \in U$ .

**Definicija 2.16** (Atlas). Atlas lokalno euklidskog prostora  $M$  je kolekcija koordinatnih karti koje prekrivaju skup  $M$ . Maksimalni atlas je atlas koji nije podskup nekog drugog atlasa.

**Definicija 2.17** (Glatki atlas). Atlas mnogostrukosti  $M$  je gladak ako za svake dvije karte mnogostrukosti  $(U \subseteq M, \varphi)$  i  $(V \subseteq M, \psi)$  vrijedi da su  $\psi \circ \varphi^{-1}$  i  $\varphi \circ \psi^{-1}$  glatke funkcije. Maksimalni glatki atlas naziva se glatka struktura, a njegove karte glatke karte.

**Definicija 2.18** (Glatka mnogostrukost). Ako mnogostrukost ima glatku strukturu, naziva se glatka mnogostrukost.

**Definicija 2.19** (Lokalne koordinate). Komponente glatke karte  $(U \subset M, \varphi)$  glatke mnogostrukosti  $M$ ,  $\varphi = (\varphi^1, \dots, \varphi^n) : M \rightarrow \mathbb{R}^n$ , nazivaju se lokalnim koordinatama mnogostrukosti  $M$ .

## 2.3 Tangentni prostor

**Definicija 2.20** (Vektor s ishodištem u točki). Vektor  $(a)_p \in V_p$  ima ishodište u točki  $p \in V$  i smjer  $a \in V$ .

**Definicija 2.21** (Reprezentativna krivulja). Za vektor  $(a)_p \in V_p$  reprezentativna krivulja je glatka funkcija  $\sigma : I \subset \mathbb{R} \rightarrow V$ , gdje je  $I$  neprazan otvoreni skup, za koju vrijedi

$$\sigma(t_0) = p \tag{2.1}$$

i

$$\frac{d\sigma}{dt}(t_0) = a. \tag{2.2}$$

**Definicija 2.22** (Usmjerena derivacija). Za funkciju  $f \in C^\infty(U \subset \mathbb{R}^n)$ ,  $f : U \rightarrow \mathbb{R}$  u točki  $p \in U$  i smjeru  $a \in \mathbb{R}^n$  usmjerena derivacija,  $D_a f(p) \in \mathbb{R}$ , definirana je relacijom

$$D_a f(p) = \frac{d}{dt}(f \circ \sigma)(t_0), \tag{2.3}$$

gdje je  $\sigma$  reprezentativna krivulja vektora  $a_p$ .

Budući da je  $\sigma$  reprezentativna krivulja, izraz 2.3 možemo zapisati pomoću komponenti vektora  $a$  kao

$$D_a f(p) = a^i \partial_i f(p), \quad (2.4)$$

gdje je korištena Einsteinova konvencija o sumaciji. Također, usmjerena derivacija nasljeđuje svojstva derivacije

**Teorem 2.1.** *Svojstva usmjerene derivacije* Za  $c \in \mathbb{R}$  i  $f, g \in C^\infty(\mathbb{R}^n)$  vrijedi

$$D_a(cf + g)(p) = cD_a f + D_a g, \quad (2.5)$$

$$D_a(fg)(p) = D_a f g + f D_a g. \quad (2.6)$$

*Dokaz.* Neka je  $\sigma$  reprezentativna krivulja vektora  $a_p$ , tada vrijedi

$$D_a(cf + g) = \frac{d}{dt}((cf + g) \circ \sigma)(p) = \frac{d}{dt}(cf \circ \sigma + g \circ \sigma)(p) = cD_a f + D_a g. \quad (2.7)$$

Analogno drugi izraz dokazujemo primjenom kompozicije

$$D_a(fg) = \frac{d}{dt}((fg) \circ \sigma)(p) = \frac{d}{dt}(f \circ \sigma g \circ \sigma)(p) \quad (2.8)$$

te dobivamo

$$D_a(fg) = g(p) \frac{d}{dt}((f) \circ \sigma)(p) + f(p) \frac{d}{dt}((g) \circ \sigma)(p) = gD_a f + fD_a g. \quad (2.9)$$

□

Koncept derivacije u smjeru možemo generalizirati na pojam derivacije. Ona je definirana zahtijevajući svojstva dana u sljedećoj definiciji

**Definicija 2.23** (Derivacija). *Derivacija je funkcija*  $X_p : C^\infty(p) \rightarrow \mathbb{R}$  *za koju vrijedi*

$$X_p(af + bg) = aX_p f(p) + bX_p g(p) \quad (2.10)$$

i

$$X_p(fg)(p) = X_p(f)g + fX_p(g), \quad (2.11)$$

gdje su  $a, b \in \mathbb{R}$  i  $f, g \in C^\infty(p)$ . Skup svih derivacija skupa  $C^\infty(p)$  označavamo oznakom  $Der(C^\infty(p))$ .

Skup  $Der(C^\infty(p))$  je vektorski prostor, što se može dokazati primjećujući da i funkcija  $O(p) = 0$  zadovoljava svojstva definicije. Kako bismo pronašli bazu prostora  $Der(C^\infty(p))$ , koristimo sljedeći teorem.

**Teorem 2.2.** *Neka je  $X_p \in Der(C^\infty(p))$ , tada postoji  $\xi \in \mathbb{R}^n$  takav da je  $X_p f = \xi^i \partial_i f$  za  $f \in C^\infty(p)$ .*

Ovaj teorem omogućava poistovjećivanje derivacija  $\partial_i$  s bazom prostora  $Der(C^\infty(p))$  jer svaki  $X_p$  možemo zapisati kao

$$X_p = \xi^i \partial_i. \quad (2.12)$$

Štoviše, za svaki vektor  $a_p \in V_p$ , možemo pronaći preslikavanje na prostor  $Der(C^\infty(p))$  pomoću reprezentativnih krivulja. Derivaciju  $X_p$  možemo zapisati pomoću komponenti vektora  $a$

$$X_p f(p) = a^i \partial_i f = D_{\dot{\sigma}(t_0)} f(\sigma(t_0)). \quad (2.13)$$

Uvedemo li preslikavanje  $T : V_p \rightarrow Der(C^\infty(p))$ , izraz 2.13 možemo zapisati kao

$$T(a_p) = D_{\dot{\sigma}(t_0)} f(\sigma(t_0)) = X_p f, \quad (2.14)$$

gdje je  $X_p \in Der(C^\infty(p))$ . Ova identifikacija omogućava definiciju tangentskog prostora

**Definicija 2.24** (Tangentni prostor). *Tangentni prostor u točki  $p \in \mathbb{R}^n$ , u oznaci  $T_p \mathbb{R}^n$  je  $n$ -dimenzionalni vektorski prostor  $Der(C^\infty(p))$ .*

Promatrajući sve točke  $p \in \mathbb{R}^n$ , dolazimo do definicije vektorskog polja  $X(x) = \xi(x)^i \partial_i$  na sljedeći način.

**Definicija 2.25** (Vektorsko polje). *Vektorsko polje  $X \in Der C^\infty(\mathbb{R})$  je funkcija  $X(p) = X_p$ , gdje je  $X_p \in T_p \mathbb{R}^n$  za svaki  $p \in \mathbb{R}^n$ .*

Prostoru  $T_p \mathbb{R}^n$  možemo pronaći dualni prostor, t.j. prostor linearnih funkcionala  $T_p \mathbb{R}^n \rightarrow \mathbb{R}$ . Ovakav linearni funkcional zovemo 1-forma, a prostor linearnih funkcionala prostora  $T_p \mathbb{R}^n$  označavamo oznakom  $T_p^* \mathbb{R}^n$ .

**Definicija 2.26** (1-forma). Funkcija  $df_p : T_p\mathbb{R}^n \rightarrow \mathbb{R}$  je 1-forma definirana relacijom  $df_p(X_p) = X_p(f)$  za  $X_p \in T_p\mathbb{R}^n$ .

Definicija 2.26 vodi do relacije koja omogućava lakše računanje s tangentnim vektorima i njihovim dualnim vektorima

**Teorem 2.3.** Vektori  $dx^i$  čine bazu na dualnom prostoru te vrijedi

$$dx^i(\partial_j) = \delta_j^i. \quad (2.15)$$

*Dokaz.* S jedne strane vrijedi

$$df(X_p) = \frac{\partial f}{\partial x^i} dx^i(\xi^j \partial_j) = \xi^j \frac{\partial f}{\partial x^i} dx^i(\partial_j), \quad (2.16)$$

gdje su  $\xi^j$  komponente vektora  $X_p$ . S druge strane vrijedi

$$X_p(f) = \xi^j \partial_j(f). \quad (2.17)$$

Izjednačavanjem dobivamo  $dx^i(\partial_j) = \delta_j^i$ . □

Možemo promatrati tangentni prostor za skup svih točaka, a ne samo za fiksni  $p$ . Budući da prostore točaka razlikujemo, radi se o svežnju prostora.

**Definicija 2.27.** Tangentni svežanj je disjunktna unija tangentnih prostora

$$T\mathbb{R}^n = \bigsqcup_{p \in \mathbb{R}^n} T_p\mathbb{R}^n,$$

a kotangentni svežanj je disjunktna unija kotangentnih prostora

$$T^*\mathbb{R}^n = \bigsqcup_{p \in \mathbb{R}^n} T_p^*\mathbb{R}^n.$$

Preslikavanje između dvaju tangentnih prostora vrši se operacijom *pushforward*, dok se preslikavanje između kotangentnih prostora vrši operacijom *pullback*.

**Definicija 2.28** (Pushforward). Za  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $\Phi \in C^\infty(\mathbb{R}^n)$ , i  $p \in \mathbb{R}^n$ , pushforward je preslikavanje  $\Phi_{*,p} : T_p\mathbb{R}^n \rightarrow T_{\Phi(p)}\mathbb{R}^m$ .

**Definicija 2.29** (Pullback). Za  $g \in T_{\Phi(p)}^*\mathbb{R}^m$ , pullback je definiran relacijom  $\Phi^*g = g \circ \Phi$ .

**Teorem 2.4.** Za  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $X_p \in T_p\mathbb{R}^n$  i za svaki  $g \in C^\infty(p)$

$$\Phi_{*,p}(X_p)g = X_p(g \circ \Phi),$$

odnosno  $\Phi_{*,p}(X_p)g \in T_{\Phi(p)}\mathbb{R}^m$ .

*Dokaz.* Definirajmo  $Y_q = \Phi_{*,p}(X_p)g$  za  $q = \Phi(p)$  i promotrimo  $g, h \in C^\infty(p)$ . Za  $a, b \in \mathbb{R}$  vrijedi linearnost

$$Y_q(ag+bh) = X_p((ag+bh) \circ \Phi) = X_p(ag \circ \Phi + bh \circ \Phi) = aX_p(g \circ \Phi) + bX_p(h \circ \Phi) = aY_q(g) + bY_q(h),$$

i pravilo deriviranja

$$Y_q(gh) = X_p((gh) \circ \Phi) = X_p((g \circ \Phi)(h \circ \Phi)) = X_p(g \circ \Phi)h \circ \Phi + g \circ \Phi X_p(h \circ \Phi) = Y_q(g)h + gY_q(h).$$

Prema tome,  $Y_q$  je derivacija i vrijedi  $Y_q \in T_{\Phi(p)}\mathbb{R}^m$ . Prema tome  $\Phi_*$  je preslikavanje sa  $T_p\mathbb{R}^n$  na  $T_{\Phi(p)}\mathbb{R}^m$  te stoga  $\Phi_*$  pushforward.  $\square$

**Teorem 2.5.** Pushforward je linearna transformacija

*Dokaz.* Promotrimo  $X_p, Y_p \in T_p\mathbb{R}^n$  i  $g \in C^\infty(p)$ . Tada vrijedi

$$\Phi_{*,p}(aX_p + Y_p)(g) = (aX_p + Y_p)(g \circ \Phi) = aX_p(g \circ \Phi) + Y_p(g \circ \Phi) = a\Phi_{*,p}(X_p) + \Phi_{*,p}(Y_p).$$

$\square$

Kao posljedica ovog teorema, vrijedi sljedeća operativna formula.

**Teorem 2.6.** Za  $X_p \in T_p\mathbb{R}^n$ ,  $X_p = \xi^i \partial_i$ ,  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$  vrijedi

$$\Phi_{*,p} = \sum_{a=1}^m \sum_{j=1}^n \xi^j \frac{\partial \Phi^a}{\partial x^j} \Big|_p \frac{\partial}{\partial y^a} \Big|_q$$

*Dokaz.* Koristeći teorem 2.4, možemo koristiti funkciju  $g \in C^\infty(p)$  za zapis

$$\Phi_{*,p}X_p(g) = \left( \sum_{i=1}^n \xi^i \frac{\partial}{\partial x^i} \right) (g \circ \Phi),$$



a zbog teorema 2.5, pushforward je linearan i vrijedi

$$\Phi_{*,p}X_p(g) = \sum_{i=1}^n \xi^i \left( \frac{\partial}{\partial x^i} \right) (g \circ \Phi).$$

Deriviranjem kompozicije dobivamo

$$\Phi_{*,p}X_p(g) = \sum_{i=1}^n \xi^i \sum_{a=1}^m \frac{\partial g}{\partial y^a} \frac{\partial \Phi^a(p)}{\partial x^i} = \left( \sum_{a=1}^m \sum_{j=1}^n \xi^j \frac{\partial \Phi^a}{\partial x^j} \Big|_p \frac{\partial}{\partial y^a} \Big|_q \right) g.$$

□

**Definicija 2.30** (Pushforward na svežnju). Za  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , pushforward  $\Phi_* : T\mathbb{R}^n \rightarrow T\mathbb{R}^m$  je definiran relacijom  $\Phi_*(X_p) = \Phi_{*,p}X_p$  za svaki  $p \in \mathbb{R}^n$  i  $X_p \in T_p\mathbb{R}^n$ .

**Teorem 2.7** (Pullback). Neka je  $Y = \eta_a dy^a(q) \in T^*\mathbb{R}^m$  vektor iz tangentskog svežnja, tada za  $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , vrijedi

$$\Phi^*Y = \sum_{a=1}^m \sum_{i=1}^n \eta_a(\Phi) \frac{\Phi^a}{\partial x^i} dx^i.$$

*Dokaz.* Dokaz je analogan dokazu teorema 2.6 ako se promatra svaka točka  $q \in \mathbb{R}^m$  zasebno. Ako  $Y_q = \eta(q)_a dy^a \in T_q^*\mathbb{R}^m$  vrijedi

$$(\Phi^*Y_q)(g) = Y_q \circ \Phi = \left( \sum_{a=1}^m \eta(q)_a dy^a \right) \circ \Phi = \sum_{a=1}^m \sum_{i=1}^n \eta_a(\Phi(p)) \frac{\partial \Phi^a(p)}{\partial x^i} dx^i$$

□

Analogno se može definirati tangentni (kotangentni) prostor za točku mnogostrukosti  $T_pM$  ( $T_p^*M$ ). Preslikavanja između mnogostrukosti omogućuju klasifikaciju preslikavanja po njegovom rangu.

**Definicija 2.31** (Konstantni rang). Neka su  $M$  i  $N$  glatke mnogostrukosti, preslikavanje  $\Phi : M \rightarrow N$  ima konstantan rang ako jakobijan  $D\Phi$  ima isti rang u svakoj točki  $p \in M$ .

**Definicija 2.32** (Submerzija). Neka su  $M$  i  $N$  glatke mnogostrukosti, tada je funkcija  $\Phi : M \rightarrow N$  submerzija ako je surjektivna u svakoj točki prostora, odnosno  $R(D\Phi) = \dim N$ .

**Definicija 2.33** (Imerzija). Neka su  $M$  i  $N$  glatke mnogostrukosti, tada je funkcija  $\Phi : M \rightarrow N$  imerzija ako je injektivna u svakoj točki prostora, odnosno  $R(D\Phi) = \dim M$ .

**Definicija 2.34** (Lokalni difeomorfizam). Neka su  $M$  i  $N$  glatke mnogostrukosti, tada je funkcija  $\Phi : M \rightarrow N$  lokalni difeomorfizam ako je i imerzija i submerzija za svaku točku  $p \in M$ .

**Teorem 2.8** (Teorem o globalnom rangu). Neka su  $M$  i  $N$  glatke mnogostrukosti, a  $\Phi : M \rightarrow N$  glatko preslikavanje konstantnog ranga. Tada vrijedi

1. Ako je  $\Phi$  surjektivna, ujedno je i glatka submerzija
2. Ako je  $\Phi$  injektivna, ujedno je i glatka imerzija
3. Ako je  $\Phi$  bijektivna, ujedno je i difeomorfizam

Definiranje operacije inkluzije omogućava klasifikaciju mnogostrukosti prema tipu preslikavanje među njima te određuje pojam podmnogostrukosti.

**Definicija 2.35** (Inkluzija). Neka je  $S$  podskup skupa  $X$ ,  $S \subseteq X$  i  $x \in S$ . Funkcija  $i_S : S \rightarrow X$  je inkluzija, u oznaci  $S \hookrightarrow X$  ako je  $i_S(x) = x$  za  $x \in S$ .

**Definicija 2.36** (Imerzija i submerzija mnogostrukosti). Neka je  $M$   $n$ -dimenzionana mnogostrukost s glatkom strukturom te vrijedi  $M \subseteq \tilde{M}$ . Mnogostrukost  $M$  se naziva

1. Imerzirana mnogostrukost, ili podmnogostrukost, ako je  $M \hookrightarrow \tilde{M}$  glatka imerzija.
2. Submerzirana mnogostrukost, ako je  $M \hookrightarrow \tilde{M}$  glatka submerzija.

Kodimenzija mnogostrukosti  $M$  je broj  $\dim \tilde{M} - \dim M$ .

## 2.4 Metrika

**Definicija 2.37** (Tenzorski produkt). Neka su  $\alpha^1, \alpha^2 \in V^*$ , tada direktni produkt  $\alpha^1 \otimes \alpha^2 \in V^* \times V^*$ ,  $\alpha^1 \otimes \alpha^2 : V \times V \rightarrow \mathbb{R}$ , ima svojstvo

$$(\alpha^1 \otimes \alpha^2)(v, w) = \alpha^1(v)\alpha^2(w). \quad (2.18)$$

**Teorem 2.9.** Direktni produkt  $\alpha^1 \otimes \alpha^2 : V \times V \rightarrow \mathbb{R}$  je bilinearna forma.

*Dokaz.* Pretpostavimo da vektore  $v$  i  $w$  možemo zapisati u bazi  $\{e_i\}$  na prostoru  $V$ ,  $v = v^i e_i$  i  $w = w^i e_i$ . Budući da su  $\alpha^1$  i  $\alpha^2$  linearni funkcionali, vrijedi

$$(\alpha^1 \otimes \alpha^2)(v, w) = (\alpha^1 \otimes \alpha^2)(v^i e_i, w^j e_j) = v^i w^j \alpha^1(e_i) \alpha^2(e_j) = v^i w^j (\alpha^1 \otimes \alpha^2)(e_i, e_j). \quad (2.19)$$

□

Metrika je bilinearna forma na tangentnom prostoru.

**Definicija 2.38** (Metrika). *Metrika  $g_p$  u točki  $p \in \mathbb{R}^n$  je bilinearna forma  $g_p : T_p \mathbb{R}^n \times T_p \mathbb{R}^n \rightarrow \mathbb{R}$ . Metrički tenzor je dan izrazom  $g_{ij} = g(\partial_i, \partial_j)$ .*

Definiciju možemo proširiti na sve točke prostora koristeći dualnu bazu. Promotrimo  $X, Y \in T\mathbb{R}^n$  za koje vrijedi

$$g(X, Y) = g(\xi^i \partial_i, \eta^j \partial_j) = \xi^i \eta^j g_{ij}. \quad (2.20)$$

Budući da vrijedi

$$\begin{aligned} dx^i(X) &= X(x^i) = \xi^j (\partial_j(x^i)) = \xi^i \\ dx^i(Y) &= Y(x^i) = \eta^j (\partial_j(x^i)) = \eta^i \end{aligned}$$

, vektore  $X$  i  $Y$  možemo zapisati kao  $X = dx^i(X) \partial_i$  i  $Y = dx^i(Y) \partial_i$  te izraz 2.20 postaje

$$g(X, Y) = g_{ij} dx^i(X) dx^j(Y). \quad (2.21)$$

Budući da su 1-forme linearni funkcionali, ovaj izraz postaje

$$g(X, Y) = g_{ij} (dx^i \otimes dx^j)(X, Y), \quad (2.22)$$

odnosno, metrika jest bilinearni funkcional na tangentnom prostoru

$$g = g_{ij} dx^i \otimes dx^j, \quad (2.23)$$

Preslikavanje između tangentnog prostora i njegovog duala možemo izvršiti operatorima  $\flat : T\mathbb{R}^n \rightarrow T^*\mathbb{R}^n$  i  $\sharp : T^*\mathbb{R}^n \rightarrow T\mathbb{R}^n$ . Njihovo djelovanje dano je sljedećim teoremom

**Teorem 2.10.** Za vektor  $X = \xi^i \partial_i$  vrijedi  $X^b = g_{\alpha i} \xi^\alpha dx^i$ , a za vektor  $X = \xi_i dx^i$  vrijedi  $X^\sharp = g^{i\alpha} \xi_\alpha \partial_i$ . Odnosno, spuštanje indeksa je operacija  $\xi_i = g_{i\alpha} \xi^\alpha$ , a podizanje  $\xi^i = g^{i\alpha} \xi_\alpha$ .

*Dokaz.* Prvu tvrdnju dokazujemo uvođenjem funkcionala  $\phi(X) \cdot = g(X, \cdot)$

$$\phi(X) = \xi_\alpha dx^\alpha. \quad (2.24)$$

Raspisujući  $X = \xi^\alpha \partial_\alpha$  te djelujući funkcionalnom  $\phi(X)$  na  $\partial_i$ , dobivamo

$$\phi(X) \partial_i = \xi^\alpha \phi(\partial_\alpha) \partial_i = \xi^\alpha g(\partial_\alpha, \partial_i) = \xi^\alpha g_{i\alpha}. \quad (2.25)$$

S druge strane raspisujemo  $\phi(X)$  u dualnoj bazi

$$\phi(X, \partial_i) = (\xi_\alpha dx^\alpha)(\partial_i) = \xi_\alpha dx^\alpha(\partial_i) = \xi_i. \quad (2.26)$$

Usporedbom vrijedi  $\xi_i = g_{i\alpha} \xi^\alpha$ . Drugu tvrdnju dokazujemo množenjem ove relacije s inverzom metrike.  $\square$

Skalarni produkt dvaju vektora,  $X$  i  $Y$ , može se zapisati na više načina

$$\langle X, Y \rangle = g(X, Y) = X^b Y. \quad (2.27)$$

Uvođenje skalarnog produkta omogućava provođenje Gram-Schmidtovog postupka. Ako je  $\{\partial_i\}$  baza na  $TM$ , ortonormirani skup vektora,  $\{Z_i\}$  dobivamo konstrukcijom ortogonalnog skupa vektora

$$Y_i = \partial_i - \sum_{j=1}^{i-1} \frac{g(\partial_i, Y_j)}{g(Y_j, Y_j)} Y_j, \quad (2.28)$$

uz  $Y_1 = \partial_1$  i normirajući  $Z_i = Y_i / \sqrt{g(Y_i, Y_i)}$ .

## 2.5 Kovarijantna derivacija

Parcijalna derivacija nije kovarijantna. Naime, pri promjeni koordinata, vektori se transformiraju kao

$$A^{i'} = \frac{\partial x^{i'}}{\partial x^j} A^j \quad (2.29)$$

$$A_{i'} = \frac{\partial x^j}{\partial x^{i'}} A_j, \quad (2.30)$$

a tada derivacija postaje

$$\partial_{\alpha'} A^{i'} = \partial_{\alpha'} \left( \frac{\partial x^{i'}}{\partial x^j} A^j \right) \neq \partial_{\alpha} A^i. \quad (2.31)$$

Uvodimo Christoffelove simbole i pomoću njih definiramo kovarijantu derivaciju.

**Definicija 2.39** (Christoffelov simbol prve vrste).

$$\Gamma_{\alpha|\mu\nu} = \frac{1}{2} (\partial_{\mu} g_{\nu\alpha} + \partial_{\nu} g_{\mu\alpha} - \partial_{\alpha} g_{\mu\nu}) \quad (2.32)$$

**Definicija 2.40** (Christoffelov simbol druge vrste).

$$\Gamma_{\mu\nu}^{\sigma} = \frac{g^{\sigma\alpha}}{2} (\partial_{\mu} g_{\nu\alpha} + \partial_{\nu} g_{\mu\alpha} - \partial_{\alpha} g_{\mu\nu}) \quad (2.33)$$

Kovarijantna derivacija se transformira kao tenzor i definirana je izrazima

**Definicija 2.41** (Kovarijantna derivacija).

$$\nabla_k A_i = \partial_k A_i - \Gamma_{ki}^{\alpha} A_{\alpha}, \quad (2.34)$$

$$\nabla_k A^i = \partial_k A^i + \Gamma_{k\alpha}^i A^{\alpha}. \quad (2.35)$$

Za vektore  $A = A^i \partial_i$  i  $B = B^i \partial_i$  vrijedi

$$\nabla_B A = (B^i \partial_i A^j + \Gamma_{ik}^j A^i B^k) \partial_j, \quad (2.36)$$

a za vektor  $A = A_i dx^i$  vrijedi

$$\nabla_B A = (B^i \partial_i A_j - \Gamma_{jk}^i A_i B^k) dx^j. \quad (2.37)$$

Kovarijantna je derivacija kompatibilna s metrikom.

**Teorem 2.11** (Kompatibilnost s metrikom).

$$\nabla_{\alpha} g_{\mu\nu} = 0. \quad (2.38)$$

*Dokaz.* Korištenjem definicije kovarijantne derivacije slijedi

$$\nabla_{\alpha} g_{\mu\nu} = \partial_{\alpha} g_{\mu\nu} - \Gamma_{\mu\alpha}^{\sigma} g_{\sigma\nu} - \Gamma_{\nu\alpha}^{\sigma} g_{\sigma\mu} = \partial_{\alpha} g_{\mu\nu} - \Gamma_{\nu|\mu\alpha} - \Gamma_{\mu|\nu\alpha}. \quad (2.39)$$

Traženi izraz dobiva se raspisujući definiciju Christoffelovih simbola prve vrste

$$\nabla_{\alpha} g_{\mu\nu} = \partial_{\alpha} g_{\mu\nu} - \frac{\partial_{\alpha} g_{\mu\nu} + \partial_{\mu} g_{\alpha\nu} - \partial_{\nu} g_{\mu\alpha} + \partial_{\alpha} g_{\mu\nu} + \partial_{\nu} g_{\mu\alpha} - \partial_{\mu} g_{\alpha\nu}}{2} = \partial_{\alpha} g_{\mu\nu} - \partial_{\alpha} g_{\mu\nu} = 0. \quad (2.40)$$

□

## 2.6 Riemannov tenzor

Riemannov tenzor definiran je sljedećom relacijom

**Definicija 2.42** (Riemannov tenzor). *Promotrimo dva vektora u prostoru parametara,  $X = x^{\mu} \partial_{\mu}$  i  $Y = y^{\mu} \partial_{\mu}$ . Riemannov tenzor je*

$$R(X, Y) = [\nabla_X, \nabla_Y] - \nabla_{[X, Y]} \quad (2.41)$$

Komponente Riemannovog tenzora mogu se izračunati po formuli

$$R_{\mu\nu\alpha\beta} = \frac{\partial_{\alpha\nu} g_{\mu\beta} + \partial_{\mu\beta} g_{\alpha\nu} - \partial_{\nu\beta} g_{\mu\alpha} - \partial_{\mu\alpha} g_{\nu\beta}}{2} + \Gamma_{\nu\alpha}^{\sigma} \Gamma_{\sigma|\mu\beta} - \Gamma_{\nu\beta}^{\sigma} \Gamma_{\sigma|\mu\alpha}. \quad (2.42)$$

## 2.7 Riccijev tenzor i skalar

Riemannov tenzor može se kontrahirati s metrikom. Ovim postupkom dobiva se Riccijev tenzor,

$$R_{\mu\nu} = g^{\alpha\sigma} R_{\alpha\mu\sigma\nu}, \quad (2.43)$$

te Riccijev skalar

$$Ric = g^{\mu\nu} R_{\mu\nu}. \quad (2.44)$$

### 3 Statistika

Prateći [9] u izvođenju prostora vjerojatnosti, u ovom poglavlju izvodimo Fisherovu metriku kao Cramer-Rao među. U poglavlju 3.3 dajemo alternativni izvod iz Kullback-Leibler divergencije prateći [2].

#### 3.1 Prostor mjere

**Definicija 3.1** ( $\sigma$ -polje). *Neka je  $\mathcal{F}$  kolekcija podprostora prostora  $\Omega$ .  $\mathcal{F}$  se naziva  $\sigma$ -polje ako i samo ako vrijedi*

1.  $\emptyset \in \mathcal{F}$
2. Za  $A \in \mathcal{F}$  vrijedi  $\Omega \setminus A \in \mathcal{F}$
3. Ako su  $A_i \in \mathcal{F}$ , tada je i  $\bigcup_i A_i \in \mathcal{F}$

Za  $\Omega$  skup i  $A \subset \Omega$ ,  $\sigma(A)$  je najmanje  $\sigma$  polje koje sadrži  $A$ . Skup  $\Omega$  i pripadno  $\sigma$ -polje nazivamo mjerljiv prostor, a funkcije između mjerljivih prostora nazivamo mjerljive funkcije, ako zadovoljavaju sljedeće svojstvo.

**Definicija 3.2** (Mjerljiva funkcija). *Neka su  $(\Omega, \mathcal{F})$  i  $(\Lambda, \mathcal{G})$  mjerljivi prostori. Funkcija  $f : \Omega \rightarrow \Lambda$  je mjerljiva funkcija ako i samo ako vrijedi  $f^{-1}(\mathcal{G}) \subset \mathcal{F}$ .*

Preslikavanje sa skupa  $\Omega$  na pozitivne brojeve nazivamo mjerom ako zadovoljava sljedeća svojstva.

**Definicija 3.3** (Prostor mjere). *Neka je  $(\Omega, \mathcal{F})$  mjerljiv prostor, funkciju  $\nu : \Omega \rightarrow \mathbb{R}$  nazivamo mjerom ako vrijedi*

1. Za svaki  $A \in \mathcal{F}$  vrijedi  $0 \leq \nu(A) \leq \infty$
2.  $\nu(\emptyset) = 0$
3. Neka su  $A_i \in \mathcal{F}$  disjunktni skupovi ( $A_i \cap A_j = \emptyset$  za  $i \neq j$ ), tada vrijedi

$$\nu \left( \bigcup_{i=1}^{\infty} A_i \right) = \sum_{i=1}^{\infty} \nu(A_i)$$

Trojku  $(\Omega, \mathcal{F}, \nu)$  nazivamo prostor mjere.

Mjerljive funkcije omogućuju induciranje mjere na mjerljivom prostoru u koji preslikavaju, ako je poznata mjera na njihovoj domeni. Ovakve mjere zovu se inducirane mjere.

**Definicija 3.4** (Inducirana mjera). *Neka je  $(\Omega, \mathcal{F}, \nu)$  prostor mjere,  $(\Lambda, \mathcal{G})$  mjerljiv prostor te  $f : \Omega \rightarrow \Lambda$  mjerljiva funkcija. Kompozicija  $\nu \circ f^{-1}$  je inducirana mjera na  $(\Lambda, \mathcal{G})$ .*

Za prostore mjere vrijedi sljedeći bitni teorem

**Teorem 3.1.** *Neka je  $(\Omega, \mathcal{F}, \nu)$  prostor mjere. Tada vrijedi*

1. *Ako je  $A \subset B \subseteq \Omega$ , tada je  $\nu(A) \leq \nu(B)$*
2. *Neka je  $A_1, \dots$  niz u  $\Omega$ . Tada vrijedi*

$$\nu \left( \bigcup_{i=1}^{\infty} A_i \right) \leq \sum_{i=1}^{\infty} \nu(A_i)$$

3. *Neka je  $A_1 \subset A_2 \subset \dots$  te  $\nu(A_1) < \infty$  tada vrijedi*

$$\nu \left( \lim_{n \rightarrow \infty} A_n \right) = \nu \left( \bigcup_{i=1}^{\infty} A_i \right) = \lim_{n \rightarrow \infty} \nu(A_n)$$

4. *Neka je  $A_1 \supset A_2 \supset \dots$  te  $\nu(A_1) < \infty$  tada vrijedi*

$$\nu \left( \lim_{n \rightarrow \infty} A_n \right) = \nu \left( \bigcap_{i=1}^{\infty} A_i \right) = \lim_{n \rightarrow \infty} \nu(A_n)$$

*Dokaz.* Primijetimo da skup  $B$  možemo zapisati kao  $B = A \cup (A^c \cap B)$  jer vrijedi  $A \cup (A^c \cap B) = (A \cup A^c) \cap (A \cup B) = \Omega \cap B = B$ . Primijetimo da također vrijedi da smo  $B$  dekomponirali na dva disjunktna dijela  $A \cap (A^c \cap B) = (A \cap A^c) \cap B = \emptyset$ . Prema tome, definicija mjere daje

$$\nu(B) = \nu(A \cup (A^c \cap B)) = \nu(A) + \nu(A^c \cap B) \geq \nu(A).$$

Drugi dio dokazujemo iterativno,  $A_2 = A_1 \cup (A_1^c \cap A_2)$ ,  $A_3 = ((A_1 \cup A_2)^c \cap A_3), \dots$  Treći dio dokazujemo uočavajući da je  $A = \lim_n A_n = \bigcup_n A_n$ . Budući da svaki sljedeći



član niza sadrži prethodni, limes od  $\nu(A_n)$  je omeđen s  $\nu(A)$ . Četvrti dio dokazujemo analogno.  $\square$

Posebni mjerljiv prostor je definiran nad skupom  $\mathbb{R}$  uvodeći  $\sigma$ -polje nad skupom  $\mathbb{R}$  - Borelovo  $\sigma$ -polje. Preslikavanje s mjerljivog prostora na  $\mathbb{R}$  nazivamo Borelovom funkcijom.

**Definicija 3.5** (Borelovo  $\sigma$ -polje). *Neka je  $\mathcal{C}$  skup svih konačnih otvorenih intervala nad  $\mathbb{R}$ . Borelovo  $\sigma$ -polje je  $\mathcal{B} = \sigma(\mathcal{C})$*

**Definicija 3.6** (Borelova funkcija). *Neka je  $(\Omega, \mathcal{F})$  mjerljiv prostor. Funkcija  $f : \Omega \rightarrow \mathbb{R}$  je Borelova funkcija ako i samo ako je  $f^{-1}(\mathcal{B}) \subset \mathcal{F}$ .*

Sljedeći korak je razviti metodu integracije. Definiramo li indikator funkciju na skupu  $\Omega$ ,

$$I_A(\omega) = \begin{cases} 1, \omega \in A \subset \Omega \\ 0, \omega \notin A \subset \Omega \end{cases}, \quad (3.1)$$

možemo jednostavno zapisati integral posebnog tipa funkcije - jednostavne funkcije, koju je moguće zapisati kao sumu indikatora.

**Definicija 3.7** (Jednostavna funkcija). *Neka su  $A_i \subset \Omega$ . Funkcija  $\varphi : \Omega \rightarrow \mathbb{R}$  je jednostavna funkcija ako se može zapisati kao*

$$\varphi(\omega) = \sum_{i=1}^k a_i I_{A_i}(\omega),$$

za  $\omega \in \Omega$  te  $a_1, \dots, a_k \in \mathbb{R}$ .

Za jednostavne funkcije definiramo integral na sljedeći način

**Definicija 3.8** (Integral jednostavne funkcije). *Neka je  $(\Omega, \mathcal{F}, \nu)$  prostor mjere i  $\varphi : \Omega \rightarrow \mathbb{R}_0^+$  nenegativna jednostavna funkcija definirana pomoću indikatora na skupovima  $A_1, \dots, A_k \subset \Omega$  i vrijednostima  $a_1, \dots, a_k$  kao u definiciji 3.7. Integral funkcije  $\varphi$  je*

$$\int_{\Omega} \varphi d\nu = \sum_{i=1}^k a_i \nu(A_i).$$

Pomoću jednostavnih funkcija moguće je napisati integral proizvoljne Borelove funkcije.

**Definicija 3.9** (Integral Borelove funkcije). *Neka je  $f$  nenegativna Borelova funkcija,  $(\Omega, \mathcal{F}, \nu)$  prostor mjere i  $\mathcal{S}_f = \{\varphi : \Omega \rightarrow \mathbb{R}_0^+ \mid \varphi(\omega) \leq f(\omega), \omega \in \Omega\}$  kolekcija svih jednostavnih funkcija koje su veće ili jednake  $f$  na skupu  $\Omega$ . Inegtal funkcije  $f$  je supremum*

$$\int f \, d\nu = \sup \left\{ \int \varphi \, d\nu \mid \varphi \in \mathcal{S}_f \right\}.$$

Za općenitu Borelovu funkciju, definiramo  $f_{\pm}(\omega) = \max(\pm f(\omega), 0)$  te vrijedi

$$\int f \, d\nu = \int f_+ \, d\nu - \int f_- \, d\nu.$$

Derivacije možemo definirati polazeći od integrala. Primijetimo da vrijedi sljedeći koji omogućava deriviranje dvaju mjera uvođenjem pojma Radon-Nikodym derivacije.

**Teorem 3.2.** *Neka je  $(\Omega, \mathcal{F}, \nu)$  prostor mjere,  $f$  nenegativna Borelova funkcija. Funkcija  $\lambda$  je također mjera na  $(\Omega, \mathcal{F})$*

$$\lambda(A) = \int_A f \, d\nu.$$

*Dokaz.* Raspišimo definiciju 3.9

$$\lambda(A) = \int f \, d\nu = \sup \left\{ \int_A \varphi \, d\nu \mid \varphi \in \mathcal{S}_f \right\}.$$

Koristeći definiciju 3.8

$$\lambda(A) = \sup \left\{ \sum_{i=1, A_i \subset A}^k a_i \nu(A_i) \mid \varphi \in \mathcal{S}_f \right\},$$

uočavamo da je  $\lambda(A) \geq 0$  jer je  $\nu(A) \geq 0$  te  $\lambda(A) \leq \infty$  jer je  $\nu(A) \leq \infty$ . Također kada je  $A = \emptyset$ ,  $\nu(A) = 0$  i  $\nu(A_i \subset \emptyset) = 0$  te je i  $\lambda(\emptyset) = 0$ . Ako su  $A_1, \dots$  disjunktni skupovi, tada je

$$\lambda \left( \bigcup_i A_i \right) = \sup \left\{ \sum_{j=1, B_j \subset \bigcup_i A_i}^k a_j \nu(B_j) \mid \varphi \in \mathcal{S}_f \right\},$$

Definiramo li  $B_{ij} = A_i \cap B_j$ , vidimo da su  $B_{ij}$  disjunktni za  $i \neq j$   $B_{ij} \cap B_{kl} = (A_i \cap$

$A_k) \cap (B_j \cap B_l) = \emptyset \cap (B_j \cap B_l) = \emptyset$ . Zbog svojstva mjere vrijedi

$$\lambda \left( \bigcup_i A_i \right) = \sup \left\{ \sum_{j=1, B_{ij} \subset \bigcup_i A_i}^k a_j \nu(B_{ij}) \mid \varphi \in \mathcal{S}_f \right\} = \sum_i \lambda(A_i),$$

odnosno  $\lambda$  je također mjera. □

**Definicija 3.10** (Radon-Nikodym derivacija). *Neka je  $(\Omega, \mathcal{F}, \nu)$  prostor mjere,  $f$  nenegativna Borelova funkcija u funkcija  $\lambda$  integral*

$$\lambda(A) = \int_{A \in \mathcal{F}} f \, d\nu.$$

*Funkcija  $f$  naziva se Radon-Nikodym derivacija, odnosno gustoća, od  $\lambda$  u odnosu na mjeru  $\nu$ , u oznaci*

$$f = \frac{d\lambda}{d\nu}.$$

### 3.2 Prostor vjerojatnosti

Uz dodatni uvjet da je  $\nu(\Omega) = 1$ , dolazimo do definicije prostora vjerojatnosti.

**Definicija 3.11** (Prostor vjerojatnosti). *Prostor mjere  $(\Omega, \mathcal{F}, P)$  naziva se prostor vjerojatnosti ako je  $P$  mjera sa svojstvom  $P(\Omega) = 1$ .*

**Definicija 3.12** (Slučajni vektor). *Neka je  $(\Omega, \mathcal{F})$  mjerljiv prostor, Borel-mjerljiva funkcija  $X : \Omega \rightarrow \mathbb{R}$  naziva se slučajna varijabla, a Borel-mjerljiva funkcija  $X : \Omega \rightarrow \mathbb{R}^k$  naziva se slučajni vektor.*

**Definicija 3.13** (Distribucija slučajne varijable). *Neka je  $(\Omega, \mathcal{F}, P)$  prostor vjerojatnosti i  $X$  slučajni vektor. Kompozicija  $P_X = P \circ X^{-1}$  naziva se distribucija od  $X$ .*

**Definicija 3.14** (Kumulativna funkcija distribucije, CDF). *Neka je  $(\mathbb{R}, \mathcal{B}, P)$  prostor vjerojatnosti. Kumulativna funkcija distribucije, CDF, je funkcija  $F(x) = P(\langle -\infty, x \rangle]$ .*

**Teorem 3.3.** *Neka je  $F$  CDF nad prostorom vjerojatnosti  $(\mathbb{R}, \mathcal{B}, P)$ . Tada vrijedi*

1.  $F(-\infty) = 0$
2.  $F(\infty) = 1$

3. za  $x \leq y$  vrijedi  $F(x) \leq F(y)$

4.  $\lim_{y \rightarrow x^+} F(y) = F(x)$

*Dokaz.* Svaki dio dokazujemo zasebno

1.  $F(-\infty) = P(\emptyset) = 0$

2.  $F(\infty) = P(\mathbb{R}) = 1$

3. Budući da vrijedi  $\langle -\infty, x \rangle \subset \langle -\infty, y \rangle$  za  $y > x$  te vrijedi teorem 3.1, vrijedi i dani rezultat.

□

**Definicija 3.15** (Gustoća vjerojatnosti, PDF). *Neka je  $(\Omega, \mathcal{F}, P)$  prostor vjerojatnosti, te  $F$  i  $P$  mjere na  $(\Omega, \mathcal{F}, P)$ . Tada nenegativnu Borelovu funkciju  $f$ ,*

$$F(A) = \int_A f \, dP,$$

odnosno Radon-Nikodym derivaciju od  $F$ ,

$$f = \frac{dF}{dP}$$

zovemo gustoćom vjerojatnosti (PDF).

Za slučajne vektore, uvodimo očekivanu vrijednost  $E$  te njihovu varijancu.

**Definicija 3.16** (Očekivana vrijednost i varijanca). *Neka je  $(\Omega, \mathcal{F}, P)$  prostor vjerojatnosti i  $X$  slučajni vektor. Očekivana vrijednost/očekivanje od  $X$ ,  $E[X]$ , je integral*

$$E[X] = \int_{\Omega} X \, dP.$$

Analogno, varijanca od  $X$  je dana izrazom  $Var[x] = E[(X - E[X])^T(X - E[X])]$ .

Za daljnja razmatranja potrebna nam je samo normalna distribucija,  $\mathcal{N}(\mu, \sigma)$ , definirana s pripadnim PDF

$$f(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{x-\mu}{\sigma}\right)^2}. \quad (3.2)$$

Njezina generalizacija u  $N$  dimenzija dana je izrazom

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi \det \Sigma)^{N/2}} e^{-(x-\mu)^T \Sigma^{-1} (x-\mu)}, \quad (3.3)$$

gdje je  $\mu \in \mathbb{R}^N$  a  $\Sigma$   $N \times N$  pozitivno-definitna matrica nazvana matricom kovarijance. Za slučajni vektor  $X \in \mu, \pm$  vrijedi  $E[X] = \mu$ ,  $Var[X] = \Sigma$ .

Kako bismo definirali  $\chi^2$  vrijednost mjerenja, bit će nam potrebna i necentralna  $\chi^2$  distribucija definirana s pripadnom PDF funkcijom za  $x > 0$

$$\chi_k^2(x) = \frac{x^{k/2-1} e^{-x/2}}{\Gamma(k/2) 2^{k/2}}. \quad (3.4)$$

Uz  $\chi_k^2$  distribuciju veže se na distribuciju od  $\chi^2 = \sum_i^k X_k^2$  u  $k$  dimenzija,  $X_i \in \mathcal{N}(\mu_i, 1)$ . Ovaj rezultat vidimo promatrajući normalnu distribuciju

$$\frac{1}{(2\pi)^{k/2}} e^{-x^2/2} dx_1 \wedge \dots \wedge dx_k = \frac{1}{(2\pi)^{k/2}} e^{-x^2/2} (\chi^2)^{k/2-1} d\chi^2 \wedge d\Omega_k.$$

Integrirajući po prostornom kutu  $d\Omega_k$ , dobivamo izraz (3.4). Za veliki  $k$ , izraz (3.4) prelazi u normalnu distribuciju.

Skup podataka je realizacija nasumičnog elementa iz prostora vjerojatnosti  $(\Omega, \mathcal{F}, P)$ , a mjeru vjerojatnosti  $P$  naziva se populacija, a same podatke uzorak. Populacija je poznata ako i samo ako je  $P(A)$  poznat za svaki  $A \in \mathcal{F}$ .

**Definicija 3.17** (Statistika). *Neka je  $X$  mjerljiva funkcija, funkcija  $T(X)$  je statistika ako je poznata kada je  $X$  poznat.*

**Teorem 3.4** (Cramer-Rao međa). *Neka je  $X = (X_1, \dots, X_n)$  uzorak iz populacije  $P$ , opisane parametrima  $\theta$  iz pripadnog prostora parametara. Neka je  $T(X)$  procjenitelj s  $E[T(X)] = g(\theta)$ , gdje je  $g$  derivabilna funkcija. Također neka  $P_\theta$  ima PDF s obzirom na mjeru  $\nu$  za sve  $\theta$  te da je  $f_\theta$  derivabilna funkcija koja zadovoljava*

$$\begin{aligned} \frac{\partial}{\partial \theta} \int f_\theta(x) d\nu &= \int \frac{\partial f_\theta(x)}{\partial \theta} d\nu, \\ \frac{\partial}{\partial \theta} \int T(x) f_\theta(x) d\nu &= \int T(x) \frac{\partial f_\theta(x)}{\partial \theta} d\nu, \end{aligned}$$

za svaki  $\theta$ . Tada vrijedi

$$\text{Var}(T(X)) \geq \left(\frac{\partial g(x)}{\partial \theta}\right)^T I^{-1} \left(\frac{\partial g(x)}{\partial \theta}\right),$$

gdje je  $I(\theta)$  Fisherova informacijska metrika

$$I(\theta)_{ij} = E \left[ \frac{\partial \ln f_\theta(X)}{\partial \theta^i} \frac{\partial \ln f_\theta(X)}{\partial \theta^j} \right]$$

*Dokaz.* Primijetimo da je  $\text{Cov}^2[T(X), X] = E^2[(T(x) - g)(X - E[X])] \leq V[T(x)]V[X]$ , što vidimo promatrajući  $E[((T[x] - g) - c(X - E[x]))^2] \geq 0$  uz  $c = \text{Cov}[T(x), x]/\text{Var}[x]$ .

Tada vrijedi

$$\text{Var}[T(x)] \geq \frac{\text{Cov}[T[X], X]}{V[X]}. \quad (3.5)$$

Računamo derivaciju u smjeru  $n^i$ ,  $D_n = n^i \partial_i$ ,

$$D_n g = \int T(x) D_n f \, dx. \quad (3.6)$$

Relacija  $D_n \int f_\theta(x) \, d\nu = 0$ , daje

$$E[D_n \ln f] = \int D_n f \, d\nu = 0, \quad (3.7)$$

te je izraz (3.6) jednak

$$D_n g = \int (T(x) - E[T(X)]) (D_n f - E[D_n f]) \, dx = \text{Cov}[T(X), D_n \ln f].$$

Također, zbog izraza (3.6) vrijedi

$$V[D_n \ln f] = E[(D_n \ln f)^2] - E^2[D_n \ln f] = E[(D_n \ln f)^2].$$

Izraz (3.5) tada postaje

$$\text{Var}[T(x)] \geq \frac{\text{Cov}[T(X), D_n \ln f]}{V[D_n \ln f]},$$

gdje je  $V[D_n \ln f] = n^i n^j I_{ij}$ . Matrični zapis daje

$$I \text{Var}[T(x)] \geq \nabla g (\nabla g)^T.$$

Množeći s lijeva s  $I^{-1}\nabla g$  dobivamo

$$\nabla g \text{Var}[T(x)] \geq \nabla g (\nabla g)^T I^{-1} \nabla g,$$

t.j.

$$\text{Var}[T(x)] \geq (\nabla g)^T I^{-1} \nabla g,$$

□

Također, vrijedi još jedno svojstvo Fisherove metrike.

**Teorem 3.5.** *Pretpostavimo da vrijedi*

$$\partial_j \int T(x) \partial_i f_\theta(x) \, d\nu = \int T(x) \partial_{ij} f \, d\nu,$$

Fisherovu metriku moguće je izraziti kao

$$I(\theta) = -E \left[ \frac{\partial^2 \ln f}{\partial \theta^i \partial \theta^j} \right].$$

*Dokaz.* Primijetimo

$$\frac{\partial \ln f}{\partial \theta^i} = \frac{-1}{f} \frac{\partial f}{\partial \theta^i} + \frac{1}{f} \frac{\partial^2 f}{\partial \theta^i \partial \theta^j}.$$

Uvjet na derivacije dovodi do iščezavanja drugog člana i preostaje traženi izraz. □

Promotrimo normalnu distribuciju u  $N$  dimenzija

$$f_\mu(x) = \frac{1}{(2\pi)^{N/2} \prod_i \sigma_i} e^{-\frac{1}{2} \sum \frac{(x^i - \mu^i)^2}{\sigma_i^2}}.$$

Distribuciju možemo normirati uvodeći rezidualne,

$$r^i = \frac{x^i - \mu^i}{\sigma_i},$$

derivirajući po rezidualima dobivamo euklidsku metriku

$$I_{ij} = \delta_{ij}. \tag{3.8}$$

### 3.3 Fisherova informacijska metrika

Za model opisan s  $N_p$  parametara  $p = (\theta^1, \dots, \theta^{N_p})$  i mjerenja,  $x = \{x_1, \dots, x_{N_d}\}$ , opisana distribucijom vjerojatnosti mjerenja  $p(x|p)$ . Ako su mjerenja nezavisna, nju možemo izraziti kao produkt vjerojatnosti svakog pojedinog  $x_i$

$$f(x|p) = \prod_{i=1}^{N_d} f(x_i|p). \quad (3.9)$$

Budući daje ovaj izraz dan produktom, korisnije je gledati log-vjerojatnost,  $\ln f(x|p)$  jer je ona aditivna po  $x_i$ . Definiramo Shannonovu entropiju izrazom

$$S(p) = - \sum_{i=1}^{N_d} f(x_i|p) \ln f(x_i|p), \quad (3.10)$$

a ako promatramo dva modela,  $p_1$  i  $p_2$ , možemo promatrati očekivanje log-vjerojatnosti modela 2 za model 1, odnosno možemo promatrati međuentropiju

$$S(p_1||p_2) = - \sum_{i=1}^{N_d} f(x_i|p_1) \ln f(x_i|p_2). \quad (3.11)$$

Razliku međuentropija modela 1 i 2 zovemo *Kullback-Leiblerovom divergencijom*,  $D_{KL}(p_1||p_2) = S(p_1||p_2) - S(p_1)$ , odnosno

$$D_{KL}(p_1||p_2) = \sum_{i=1}^{N_d} f(x_i|p_1) \ln \frac{f(x_i|p_1)}{f(x_i|p_2)}. \quad (3.12)$$

**Teorem 3.6** (Svojstva KL divergencije). *Za KL divergenciju vrijedi:*

1. *K-L divergencija je nenegativna,*

$$D_{KL}(p_1||p_2) \geq 0$$

2.  *$D_{KL}(p_1||p_2) = 0$  ako i samo ako je  $p_1 = p_2$*

3. *K-L divergencija nije simetrična,*

$$D_{KL}(p_1||p_2) \neq D_{KL}(p_2||p_1)$$

4. *K-L divergencija ne zadovoljava relaciju trokuta*



*Dokaz.* 1. Prvo svojstvo dokazujemo primjećujući da za svaki  $x > 0$  vrijedi  $\ln x \leq (x - 1)$ , stoga vrijedi nejednakost

$$D_{KL}(p_1||p_2) = - \sum_{i=1}^{N_d} f(x_i|p_1) \ln \frac{f(x_i|p_2)}{f(x_i|p_1)} \geq - \sum_{i=1}^{N_d} f(x_i|p_1) \left( \frac{f(x_i|p_2)}{f(x_i|p_1)} - 1 \right).$$

Raspisom dobivamo

$$D_{KL}(p_1||p_2) \geq \sum_{i=1}^{N_d} f(x_i|p_2) - f(x_i|p_1) = -(1 - 1) = 0.$$

2. Za  $p_1 = p_2$  vrijedi  $D_{KL}(p_1||p_1) = 0$ . Obrat računamo direktno

$$D_{KL}(p_1||p_2) = 0 \Rightarrow \ln f(x_i|p_1) - \ln f(x_i|p_2) = 0 \Rightarrow f(x_i|p_1) = f(x_i|p_2) \Rightarrow p_1 = p_2.$$

3. Promotrimo razliku

$$|D_{KL}(p_1||p_2) - D_{KL}(p_2||p_1)|.$$

Tada vrijedi

$$\begin{aligned} |D_{KL}(p_1||p_2) - D_{KL}(p_2||p_1)| &= \left| \sum_{i=1}^{N_d} (f(x_i|p_1) + f(x_i|p_2)) \ln \frac{f(x_i|p_1)}{f(x_i|p_2)} \right| \\ &> \left| \sum_{i=1}^{N_d} (f(x_i|p_1) + f(x_i|p_2)) \left( \frac{f(x_i|p_2)}{f(x_i|p_1)} - 1 \right) \right| \\ &= \left| 1 - \sum_{i=1}^{N_d} \frac{f(x_i|p_2)^2}{f(x_i|p_1)} \right|. \end{aligned}$$

4. Promotrimo razliku

$$\begin{aligned} &D_{KL}(p_1||p_2) + D_{KL}(p_2||p_3) - D_{KL}(p_1||p_3) \\ &= S(p_1||p_2) - S(p_1) + S(p_2||p_3) - S(p_2) - S(p_1||p_3) + S(p_1) \\ &= S(p_1||p_2) + S(p_2||p_3) - S(p_1||p_3) - S(p_2) \end{aligned}$$

Uzmimo slučaj samo jednog mjerenja,  $x_1$ , i odaberimo poredak  $p$  tako da vrijedi  $f(x_1|p_1) < f(x_1|p_2) < f(x_1|p_3)$ . Ovaj poredak povlači  $S(p_1||p_2) < S(p_1)$  i

$S(p_2||p_3) < S(p_2)$  te vrijedi

$$D_{KL}(p_1||p_2) + D_{KL}(p_2||p_3) - D_{KL}(p_1||p_3) < S(p_1) + S(p_2) - S(p_1||p_3) - S(p_2).$$

Ovaj izraz jednak manji je od nule

$$D_{KL}(p_1||p_2) + D_{KL}(p_2||p_3) - D_{KL}(p_1||p_3) < -D_{KL}(p_1||p_3) < 0.$$

odnosno pronašli smo slučaj kada ne vrijedi relacija trokuta,

$$D_{KL}(p_1||p_2) + D_{KL}(p_2||p_3) \geq D_{KL}(p_1||p_3).$$

□

Budući da KL-divergencija nije simetrična na zamjenu modela te ne zadovoljava nejednakost trokuta, stoga ne može biti metrika na prostoru parametara. Međutim, ako promotrimo samo mala odstupanja  $\theta_2^\alpha = \theta_1^\alpha + \Delta\theta^\alpha$  od minimuma vjerojatnosti,

$$D_{KL}(p_1||p_2) = D_{KL}(p_1||p_1) + D_{KL,\alpha}\Delta\theta^\alpha + \frac{1}{2}D_{KL,\alpha\beta}\Delta\theta^\alpha\Delta\theta^\beta + O(\Delta\theta^3), \quad (3.13)$$

gdje smo koristili pokratu  $\partial/\partial\theta^\alpha = ,_\alpha$ . Prvi član,  $D_{KL}(p_1||p_1)$ , jednak je nuli, a drugi član je očekivanje derivacije log-vjerojatnosti

$$D_{KL,\alpha} = \sum_{i=1}^{N_d} f(x_i|p_1)(\ln f(x_i|p_2))_{,\alpha} = E[(\ln f)_{,\alpha}]. \quad (3.14)$$

Budući da je minimum u  $p_1$ , i očekivanje ove logaritamske derivacije je nula

$$E[(\ln f)_{,\alpha}] = 0. \quad (3.15)$$

Ostao je samo član

$$g_{\alpha\beta} = -\frac{1}{2} \sum_{i=1}^{N_d} f(x_i|p_1)(\ln f(x_i|p_2))_{,\alpha\beta}, \quad (3.16)$$

a on je ekvivalentan Fisherovoj metrici po teoremu 3.5.

## 4 Prostor mjerenja i prostor parametara

Skup od  $N_d$  mjerenja možemo promatrati kao skup od  $N_d$  točaka  $\{y^i\}_{i=1,\dots,N_d}$  koje definiraju  $N_d$ -dimenzionalni vektorski prostor,  $\mathcal{M}$ . Takav vektorski prostor ima bazu,  $\{\partial_{y^i}\}_{i=1,\dots,N_d}$ , te odgovarajuću dualnu bazu  $\{dy^i\}_{i=1,\dots,N_d}$ . Skup mjerenja opisujemo vektorom u  $N_d$ -dimenzionalnom vektorskom prostoru

$$\mathbf{y} = y^i \partial_i. \quad (4.1)$$

S druge strane, model  $f(\theta)$  opisujemo skupom od  $N_p$  parametara  $\{\theta^\mu\}_{\mu=1,\dots,N_p}$ . Koristimo grčka slova kako bi se lakše razlikovali indeksi prostora mjerenja od prostora parametara te jako otisnute oznake za vektore iz prostora mjerenja. Definiramo vektor u  $N_p$ -dimenzionalnom prostoru parametara,  $\mathcal{P}$ ,

$$\theta = \theta^\mu \partial_\mu. \quad (4.2)$$

Model izvrjednjujemo za svaki  $y^i$ , i dobivamo vektor u prostoru parametara koji opisuje model opisan parametrima  $\theta$

$$\mathbf{f}(\theta) = f^i(\theta) \partial_i. \quad (4.3)$$

Funkciju  $\chi^2$  možemo zapisati pomoću reziduala

$$\mathbf{r} = \frac{y^i - f^i(\theta)}{\sigma_i} \partial_i, \quad (4.4)$$

gdje je  $\sigma_i$  pogreška pridružena  $i$ -tom mjerenju, na sljedeći način

$$\chi^2(\theta) = \mathbf{r}(\theta)^b \mathbf{r}(\theta) = r_i(\theta) r^j(\theta) dy^i \partial_j, \quad (4.5)$$

gdje smo pretpostavili da je prostor mjerenja euklidski,  $\mathbf{r}^b = r_i dy^i = r^i dy^i$ . Koristeći teorem 2.3, dolazimo do standardnog izraza za  $\chi^2$ ,

$$\chi^2 = \sum_{i=1}^{N_d} r_i^2. \quad (4.6)$$

U slučaju da su reziduali temeljeni na normalnoj distribuciji,  $\mathcal{N}(0, 1)$ , raspodjela vjerojatnosti u prostoru parametara može se zapisati kao Gaussijan

$$P(\mathbf{r}, \theta) dr^1 \wedge dr^2 \wedge \dots \wedge dr^{N_d} = \frac{1}{(2\pi)^{N_d/2}} e^{-\frac{\mathbf{r}(\theta)^\flat \mathbf{r}(\theta)}{2}} dr^1 \wedge dr^2 \wedge \dots \wedge dr^{N_d}, \quad (4.7)$$

gdje je distribucija  $P$  normirana u prostoru mjerenja

$$\int P(\mathbf{r}, \theta) dr^1 \wedge dr^2 \wedge \dots \wedge dr^{N_d} = 1, \quad (4.8)$$

za svaki  $\theta = \theta^\mu \partial_\mu$ .

U prostoru parametara, udaljenost između dvije točke možemo pronaći promatrajući promjene reziduala usljed male promjene parametara. Ako se u prostoru parametara pomaknemo za  $d\theta^\mu$ , rezidual se promijeni za iznos

$$d\mathbf{r} = \partial_\mu \mathbf{r} d\theta^\mu = \partial_\mu r^i d\theta^\mu \otimes \partial_i. \quad (4.9)$$

Možemo definirati metriku na prostoru parametara koristeći pullback pomoću reziduala

$$\mathbf{r}^* g = d\mathbf{r}^\flat d\mathbf{r} = \partial_\mu \mathbf{r}^\flat \partial_\nu \mathbf{r} d\theta^\mu \otimes d\theta^\nu = \sum_{i=1}^{N_d} \partial_\mu r^i \partial_\nu r^i d\theta^\mu \otimes d\theta^\nu, \quad (4.10)$$

gdje smo operatorom  $\flat$  spustili indeks u prostoru mjerenja. Drugim riječima, promatrajući rezidual kao funkciju imerzije, mnogostrukost  $\mathbf{r}(\mathcal{P})$  je podmnožica mnogostrukosti nad prostorom mjerenja,  $\mathbf{r}(\mathcal{P}) \hookrightarrow \mathcal{M}$ , te ima dimenziju  $N_p$  i kodimenziju  $N_d - N_p$ .

Izraz 4.10 definira metrički tenzor

$$g_{\mu\nu} = \sum_{i=1}^{N_d} \partial_\mu r^i \partial_\nu r^i, \quad (4.11)$$

što omogućava određivanje skalarnog produkta u prostoru parametara, primjerice između dva parametra,  $\theta_1$  i  $\theta_2$  kao

$$g(\theta_1, \theta_2) = g_{\mu\nu} \theta_1^\mu \theta_2^\nu. \quad (4.12)$$

Također, kvadrat norme vektora  $\theta$  u prostoru parametara je tada

$$g(\theta, \theta) = g_{\mu\nu}\theta^\mu\theta^\nu. \quad (4.13)$$

Notaciju iz izraza 4.11 možemo pojednostaviti uvođenjem Jacobijana

$$\mathbf{J}_\mu(\theta) = \partial_\mu \mathbf{r} = \partial_\mu r^i(\theta)\partial_i, \quad (4.14)$$

dobivajući izraz za metrički tenzor

$$g_{\mu\nu}(\theta) = \mathbf{J}_\mu^\flat \mathbf{J}_\nu. \quad (4.15)$$

Metrika čije su komponente definirane izrazom 4.15 zove se Fisherova metrika.

## 4.1 Optimalno računanje Christoffelovih simbola

Budući da rješavanje geodezijske jednadžbe značajno opterećuje računalne resurse, prvo moramo naći optimalni način računanja Christoffelovih simbola,  $\Gamma_{\mu\nu}^\beta$ , za metriku danu jednadžbom 4.15. Definijski izraz za Christoffelove simbole glasi

$$\Gamma_{\mu\nu}^\beta = \frac{1}{2}g^{\alpha\beta} (\partial_\nu g_{\mu\alpha} + \partial_\mu g_{\nu\alpha} - \partial_\alpha g_{\mu\nu}). \quad (4.16)$$

Ovaj generalni izraz možemo pojednostaviti koristeći izraz 4.15 i notaciju prethodnog potpoglavlja

$$\Gamma_{\mu\nu}^\beta = \frac{1}{2}g^{\alpha\beta} \left( \partial_\nu (\mathbf{J}_\alpha^\flat \mathbf{J}_\mu) + \partial_\mu (\mathbf{J}_\alpha^\flat \mathbf{J}_\nu) - \partial_\alpha (\mathbf{J}_\mu^\flat \mathbf{J}_\nu) \right). \quad (4.17)$$

Uvodeći hesijan,  $\mathbf{H}_{\mu\nu} = \partial_{\mu\nu} \mathbf{r}$ , izraz 4.17 postaje

$$\Gamma_{\mu\nu}^\beta = \frac{1}{2}g^{\alpha\beta} \left( \mathbf{H}_{\alpha\nu}^\flat \mathbf{J}_\mu + \mathbf{J}_\alpha^\flat \mathbf{H}_{\mu\nu} + \mathbf{H}_{\alpha\mu}^\flat \mathbf{J}_\nu + \mathbf{J}_\alpha^\flat \mathbf{H}_{\mu\nu} - \mathbf{H}_{\alpha\mu}^\flat \mathbf{J}_\nu - \mathbf{J}_\mu^\flat \mathbf{H}_{\alpha\nu} \right), \quad (4.18)$$

iz čega slijedi

$$\Gamma_{\mu\nu}^\beta = g^{\alpha\beta} \mathbf{J}_\alpha^\flat \mathbf{H}_{\mu\nu}. \quad (4.19)$$

Budući da ovdje izvedeni oblik Fisherove metrike sadrži Jakobijan i Hesijan valja primijetiti da je on jednostavniji za računalnu implementaciju od sekvencijalne primjene

prvih derivacija koje bi se računale ako bismo koristili definicijski oblik Christoffelovih simbola dan izrazom 4.16.

## 4.2 Aproksimacijska metoda mnogostrukosti s rubom

Redukciju modela možemo provesti promatrajući ponašanje modela daleko od točke najbolje prilagodbe. Aproksimacijska metoda mnogostrukosti s rubom (MBAM) je metoda kojom je moguće reducirati komplicirani model pomoću informacija o metrici na prostoru parametara.

MBAM metoda se sastoji od proračuna geodezijske jednadžbe

$$\ddot{\theta}^\mu = -\Gamma_{\alpha\beta}^\mu \dot{\theta}^\alpha \dot{\theta}^\beta,$$

počevši iz točke najbolje prilagodbe u smjeru svojstvenog vektora koji odgovara najmanjoj svojstvenoj vrijednosti Fisherove metrike. Ovaj smjer je ujedno smjer najveće neodređenosti jer se KL divergencija najmanje mijenja u tome smjeru. Ovu činjenicu vidimo promatrajući razvoj 3.13. Odaberemo li pomak u bazi svojstvenih vektora,  $\Delta\theta = \sum_i \alpha_i v_i$ , koji odgovaraju svojstvenim vrijednostima Fisherove metrike,  $\lambda_i$ , izraz 3.13 postaje

$$D_{KL}(p_1||p_2) = D_{KL}(p_1||p_1) + \frac{1}{2}g \left( \sum_i \alpha_i v_i, \sum_i \alpha_i v_i \right) = D_{KL}(p_1||p_1) + \frac{1}{2} \sum_i \lambda_i \alpha_i^2.$$

Vidimo da će se KL divergencija najmanje promijeniti za pomak u smjeru najmanjeg  $\lambda_i$ .

## 4.3 Zakrivljenost

### 4.3.1 Intrinzična zakrivljenost

Koristeći izraz 2.42 i definiciju Jakobijana i Hesijana, dobivamo

$$R_{\mu\nu\alpha\beta} = 2 \left( \mathbf{H}_{\mu[\alpha}^b \mathbf{H}_{\beta]\nu} + g^{\rho\sigma} \mathbf{J}_\rho^b \mathbf{H}_{\nu[\alpha} \mathbf{H}_{\beta]\mu}^b \mathbf{J}_\sigma \right). \quad (4.20)$$

Ovaj se izraz može drugačije zapisati kao

$$R_{\mu\nu\alpha\beta} = 2\mathbf{H}_{\mu[\alpha}^b \left(1 - \mathbf{H}_{[\alpha\mu}^b g^{\rho\sigma} \mathbf{J}_\rho \mathbf{J}_\sigma^b\right) \mathbf{H}_{\beta]\nu}. \quad (4.21)$$

Uvođenjem oznake za projektor

$$\mathbf{P} = \mathbf{1} - g^{\rho\sigma} \mathbf{J}_\rho \otimes \mathbf{J}_\sigma^b, \quad (4.22)$$

ovaj izraz postaje

$$R_{\mu\nu\alpha\beta} = 2\mathbf{H}_{\mu[\alpha}^b \mathbf{P} \mathbf{H}_{\beta]\nu}. \quad (4.23)$$

te pripadni Riccijev tenzor i skalar

$$R_{\mu\nu} = \mathbf{H}^b \mathbf{P} \mathbf{H}_{\mu\nu} - \mathbf{H}_{\nu\sigma}^b \mathbf{P} \mathbf{H}_\mu^\sigma, \quad (4.24)$$

$$R = \mathbf{H}^b \mathbf{P} \mathbf{H} - \mathbf{H}_{\mu\sigma}^b \mathbf{P} \mathbf{H}^{\mu\sigma}, \quad (4.25)$$

gdje je korištena pokrata  $\mathbf{H} = \mathbf{H}_\mu^\mu$ .

Operator  $\mathbf{P}$  jest projektor, jer ga možemo konstruirati Gram-Schmidtovim postupkom. Vektore koji pripadaju tangentnom svežnju  $Tr(\mathcal{M})$  konstruiramo operacijom pushforward

$$\mathbf{X}_i = \mathbf{r}^* \partial_\mu = J^{\mu i} \partial_i.$$

Ovaj skup vektora je ortogonalan

$$g(\mathbf{X}_i, \mathbf{X}_j) = J^{\mu i} J^{\nu j} g(\partial_\mu, \partial_\nu) = \delta^{ij}$$

Promotrimo (normirani) skup vektora na prostoru mjerenja  $\{\partial_i\}$ . Od svakog od ovih vektora možemo oduzeti projekcije na vektore,  $\mathbf{X}_i$ ,

$$\mathbf{Y}_i = \partial_i - \sum_{j=1}^{N_p} g(\partial_i, \mathbf{X}_j) \mathbf{X}_j = \partial_i - \sum_{j=1}^{N_p} J^{k\mu} J^{j\nu} g(\partial_i, \partial_k) \partial_j = \partial_i - \sum_{j=1}^{N_p} J_\mu^i J_\nu^j g^{\mu\nu} \partial_j.$$

Ovaj izraz možemo zapisati matrično

$$\mathbf{Y}_i = \sum_j \left( \delta^{ij} - J_\mu^i J_\nu^j g^{\mu\nu} \right) \partial_j = \sum_j P^{ij} \partial_j,$$

gdje je  $P^{ij}$  projektor,  $P^{ij} = \delta^{ij} - J_\mu^i J_\nu^j g^{\mu\nu}$ . Da je operator  $P^{ij}$  projektor, vidimo direktnim računom

$$P^{ij} P_j^k = (\delta^{ij} - J^{i\mu} J_\mu^j)(\delta_j^k - J_j^\nu J_\nu^k) = \delta^{ik} + J^{i\mu} J_\mu^j J_j^\nu J_\nu^k - 2J^{i\mu} J_\mu^k = \delta^{ik} - J^{i\mu} J_\mu^k = P^{ik},$$

gdje smo koristili činjenicu da je  $J_\mu^j J_j^\nu = g_\mu^\nu$ . Ortonormiranu bazu možemo konstruirati iz skupa  $\{\mathbf{Y}_i\}$  koristeći Gram-Schmidtov postupak.

Ukratko, konstruirali smo bazu tangentskog svežnja prostoru parametara,  $T\mathcal{P}$ , njegovu imerziju na prostor mjerenja,  $\mathbf{X}_1, \dots, \mathbf{X}_{N_p} \in Tr(\mathcal{P}) \subseteq T\mathcal{M}$  i bazu normalnog svežnja,  $Nr(\mathcal{P}) \subseteq T\mathcal{M}$ , koristeći operator projekcije.

### 4.3.2 Ekstrinzična zakrivljenost

Promotrimo krivulju u prostoru parametara definiranu parametrom  $\tau$ , t.j.  $\theta(\tau) = \theta^\mu(\tau)\partial_\mu$ . Budući da rezidual ovisi parametarski o  $\theta$ , rezidual je također funkcija od  $\tau$

$$\mathbf{r}(\tau) = (\mathbf{r} \circ \theta)(\tau). \quad (4.26)$$

Tangentni vektor koji odgovara putanji,  $(\mathbf{v})_{\mathbf{r}(\tau)}$ , jest

$$\mathbf{v} = D_{\dot{\theta}} \mathbf{r} = \dot{\theta}^\mu \partial_\mu \mathbf{r}. \quad (4.27)$$

Analogno brzini  $\mathbf{v}$ , možemo definirati i akceleraciju  $\mathbf{a} = D_{\dot{\theta}} \mathbf{v}$ ,

$$\mathbf{a} = \ddot{\theta}^\mu \mathbf{J}_\mu + \dot{\theta}^\mu \dot{\theta}^\nu \mathbf{H}_{\mu\nu}, \quad (4.28)$$

a koristeći definiciju projektora i geodezijsku jednadžbu, dobivamo izraz

$$\mathbf{a} = \dot{\theta}^\mu \dot{\theta}^\nu \mathbf{P} \mathbf{H}_{\mu\nu}. \quad (4.29)$$

Izrazi  $\mathbf{a}$  i  $\mathbf{v}$  možemo iskoristiti za definiranje centripetalnog radijusa

$$R = \frac{\mathbf{v}^b \mathbf{v}}{\sqrt{\mathbf{a}^b \mathbf{a}}}. \quad (4.30)$$



Ovaj se izraz može pojednostaviti koristeći

$$|\mathbf{v}|^2 = \mathbf{v}^b \mathbf{v} = \dot{\theta}^\mu \dot{\theta}^\nu \mathbf{J}^b \mathbf{J} = \dot{\theta}^\mu \dot{\theta}_\mu, \quad (4.31)$$

te

$$\mathbf{a} = \dot{\theta}^\mu \dot{\theta}^\nu \dot{\theta}^\alpha \dot{\theta}^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P}^b \mathbf{P} \mathbf{H}_{\alpha\beta} = \dot{\theta}^\mu \dot{\theta}^\nu \dot{\theta}^\alpha \dot{\theta}^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta}, \quad (4.32)$$

gdje smo koristili činjenicu da je  $P$  projektor. U konačnici dobivamo

$$R(\dot{\theta}) = \frac{\dot{\theta}^\mu \dot{\theta}_\mu}{\sqrt{|\dot{\theta}^\mu \dot{\theta}^\nu \dot{\theta}^\alpha \dot{\theta}^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta}|}}. \quad (4.33)$$

Također možemo konstruirati veličinu analognu kutnoj brzini,  $\omega = |v|/R$

$$\omega(\dot{\theta}) = \sqrt{\frac{\dot{\theta}^\mu \dot{\theta}^\nu \dot{\theta}^\alpha \dot{\theta}^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta}}{\dot{\theta}^\mu \dot{\theta}_\mu}}. \quad (4.34)$$

Uvedemo li normu brzine,  $\dot{\theta} = |\mathbf{v}|n$ , gdje je  $n^\mu n_\mu = 1$ , vidimo da kutna brzina zasebno ovisi o iznosu brzine i smjeru  $n$

$$\omega(\dot{\theta}) = |\mathbf{v}| \sqrt{|n^\mu n^\nu n^\alpha n^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta}|}. \quad (4.35)$$

Uvedemo li veličinu  $\Omega(n) = \sqrt{|n^\mu n^\nu n^\alpha n^\beta \mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta}|}$ , vidimo da možemo zasebno proučavati kako se  $\omega$  mijenja sa smjerom brzine  $n$  te s njezinim iznosom

$$\omega(\dot{\theta}) = |\mathbf{v}| \Omega(n). \quad (4.36)$$

## 5 Računalna implementacija

### 5.1 Algoritamska diferencijacija

Svaku složenu funkciju možemo predstaviti kao kompoziciju jednostavnih funkcija. Algoritamska diferencijacija primjenjuje računanje derivacija jednostavnih funkcija kako bi se izračunala derivacija kompozicija. Algoritam dekomponira funkciju  $f : \mathbb{R}^M \rightarrow \mathbb{R}^N$  u niz funkcija  $a_i : \mathbb{R}^{N_{i-1}} \rightarrow \mathbb{R}^{N_i}$ ,  $i \in \{1, \dots, k\}$  kao

$$f = \bigcirc_{i=1}^k a_i = a_k \circ \dots \circ a_1. \quad (5.1)$$

Svakoj funkciji  $a_i$  je poznat njezin Jacobijan, tako da je derivaciju funkcije  $f(x)$ ,  $x \in \mathbb{R}^M$ , jednostavno odrediti primjenjujući pravilo derivacije kompozicije. Koristeći zapis po komponentama Jacobijan  $\alpha$ -te komponente funkcije  $f$ , po  $\beta$ -toj komponenti,  $x^\beta$ , je

$$\frac{\partial f^\alpha}{\partial x^\beta} = \frac{\partial f^\alpha}{\partial a_k^{\alpha_k}} \prod_{i=2}^k \frac{\partial a_i^{\alpha_i}}{\partial a_{i-1}^{\alpha_{i-1}}} \frac{\partial a_1^{\alpha_1}}{\partial x^\beta}. \quad (5.2)$$

Iako je matematički postupak kojime računamo produkt irelevantan, postoji razlika u kompleksnosti. Ako je  $M > N$ , računajući unaprijed  $(N_1 \times M, N_2 \times (N_1 \times M), \dots)$  računalo mora više uzastopnih operacija izvršiti s matricama većih dimenzija nego u slučaju računanja unatrag  $(N \times N_k, (N \times N_k) \times N_{k-1}, \dots)$ .

Koristili smo računalnu implementaciju algoritamske derivacije u paketu `autograd`<sup>2</sup>. `Autograd` je python implementacija reverzne algoritamske diferencijacije koristeći `overload` operatora poput zbrajanja i množenja po elementima kako bi omogućio deriviranje funkcija prisutnih u paketu `numpy`.

---

<sup>2</sup><https://github.com/HIPS/autograd>

## 5.2 Analitički rješivi primjeri

Najjednostavniji model je linearni model,

$$f(t, \theta) = \theta^0 + t\theta^1, \quad (5.3)$$

opisan s dva parametra  $\theta^0, \theta^1 \in \mathbb{R}$ . Ovaj model je mnogostrukost bez ruba jer je homeomorfan  $\mathbb{R}^2$ . Fisherova metrika za linearni model je konstantna

$$g = \begin{pmatrix} \Sigma & \Sigma_1 \\ \Sigma_1 & \Sigma_2 \end{pmatrix}, \quad (5.4)$$

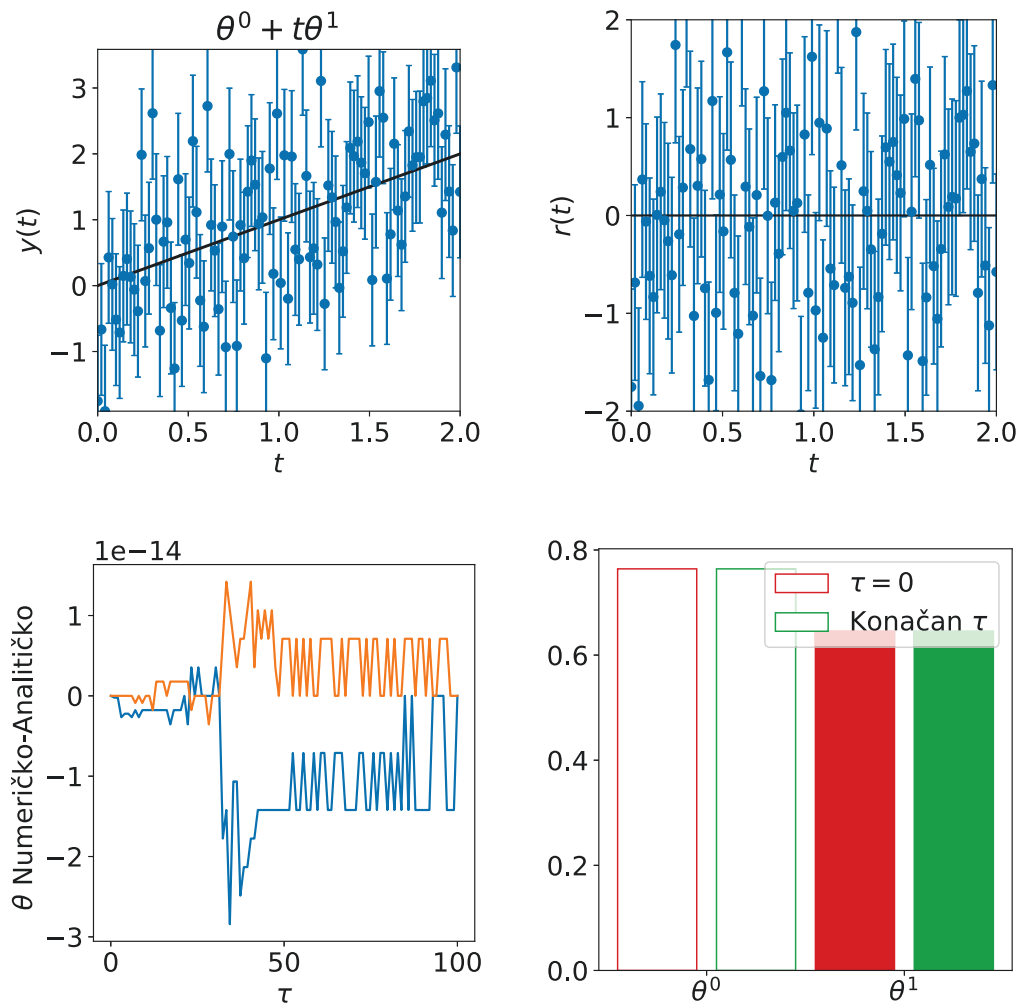
gdje smo koristili pokratu

$$\Sigma_k = \sum_{i=1}^{N_d} \frac{t_i^k}{\sigma_i^2}.$$

Budući da je metrika konstantna i Jakobijan i Hesijan za linearni model jednak 0 te su geodezici ravne linije

$$\theta^i(\tau) = \theta_{bf}^i + \dot{\theta}_{bf}^i \tau.$$

Na slici 5.1 dan je prikaz simuliranog uzorka od 100 mjerenja linearnog modela sa  $\sigma_i = 1$ . Na okvirima su, s lijeva na desno, prikazani: mjerenja, pripadni reziduali, prostor parametara s  $\chi^2$  vrijednostima i smjerovima najmanjeg svojstvenog vektora te usporedba analitičkog i numeričkog rješenja  $\theta^\mu(\tau)$ . Vidimo dobro slaganje numeričkog i analitičkog rješenja u ovakvom jednostavnom modelu. Na lijevom je okviru crnom bojom prikazana prilagodba u točki najbolje prilagodbe.



Slika 5.1: Linearan model. Okviri s lijeva na desno: Okvir 1.: Prikaz simuliranih 100 mjerenja te model najbolje prilagodbe (crna). Okvir 2.: pripadni rezidualni modela. Okvir 3.: Prostor parametara s izvrijednjenim  $\chi^2$  vrijednostima, crvenom točkom koja označava točku najbolje prilagodbe, crvenom linijom koja prikazuje izračunati geodezik te svojstvenim vektorom koji odgovara najmanjoj svojstvenoj vrijednosti metrike prikazan strelicama za različite parametre. Okvir 4.: Histogram razlika analitičkog i numeričkog rješenja za sve izračunate vrijednosti  $\tau$  za  $\theta^0$  (plava) i  $\theta^1$  (narančasta).

Kompliciraniji primjer je model potencije

$$f(t, \theta) = \theta^0 + \frac{(t\theta^1)^n}{n},$$

gdje je  $\theta^0 \in \mathbb{R}$  i  $\theta^1 \in \mathbb{R}_0^+$ . Vidimo da ovaj prostor parametara predstavlja mnogostrukost s rubom jer je homeomorfan prostoru  $\mathbb{R} \times \mathbb{H}^1$ . Metrika za ovakav model dana je relacijom

$$g_{\mu\nu} = \begin{pmatrix} \Sigma & \Sigma_n(\theta^1)^{n-1} \\ \Sigma_n(\theta^1)^{n-1} & \Sigma_{2n}(\theta^1)^{2(n-1)} \end{pmatrix}_{\mu\nu},$$

a Christoffelov simbol druge vrste je

$$\Gamma_{\alpha\beta}^{\mu} = \frac{n-1}{\theta^1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\mu} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}_{\alpha\beta}.$$

Ovaj model je analitički rješiv i svodi se na dvije razvezane diferencijalne jednačbe

$$\begin{aligned} \ddot{\theta}^0(\tau) &= 0 \\ \ddot{\theta}^1 &= -\frac{n-1}{\theta^1}(\dot{\theta}^1)^2. \end{aligned}$$

Rješenje ovog sustava je

$$\begin{aligned} \theta^0(\tau) &= \theta_{bf}^0 + \dot{\theta}^0 \tau \\ \theta^1(\tau) &= \left(\theta_{bf}^1\right)^{1-1/n} \left(\theta_{bf}^1 + n\dot{\theta}^1 \tau\right)^{1+1/n}. \end{aligned}$$

Primijetimo da će model doći do ruba mnogostrukosti,  $\theta^1 = 0$  u slučaju kada je  $\dot{\theta}_{bf}^1 < 0$  u točki najbolje prilagodbe za

$$\tau_{max} = \frac{\theta_{bf}^1}{|\dot{\theta}_{bf}^1|n}.$$

Za taj  $\tau$ , metrika više nema rang 2, već postaje signularna,

$$g_{\mu\nu}(\tau_{max}) = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix}_{\mu\nu},$$

te će se odeint integracija prekinuti.

Na slikama 5.2-5.4 prikazani su rezultati primjene MBAM metode na 100 simuliranih mjerenja kvadratnog modela. Slika 5.2 prikazuje situaciju kada je model izvrijednjen za  $t \in [0, 2]$  te svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti teži ka  $\tau^0 = -\infty$ ,  $\tau^1 = \infty$ . S druge strane, slika 5.3 prikazuje isti model, ali izvrijednjen za  $t \in [0, 1]$ , dok su na slici 5.4 prikazane usporedbe svojstvenih vektora na početku i kraju geodezika. U slučaju  $t \in [0, 1]$ , svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti dovodi do ruba mnogostrukosti za  $\tau_{max}$ . Prema tome, u ovom slučaju MBAM metoda ukazuje na izbor  $\theta^1 = 0$  kao redukciju modela, što se vidi i s desnog okvira slike 5.4, gdje je dominantna komponenta svojstvenog vektora koji odgovara najmanjoj svojstvenoj vrijednosti na rubu geodezijske krivulje. U slučaju,  $t \in [0, 2]$ , lijevi okvir slike 5.4 ukazuje na moguću eliminaciju  $\theta^0$ .

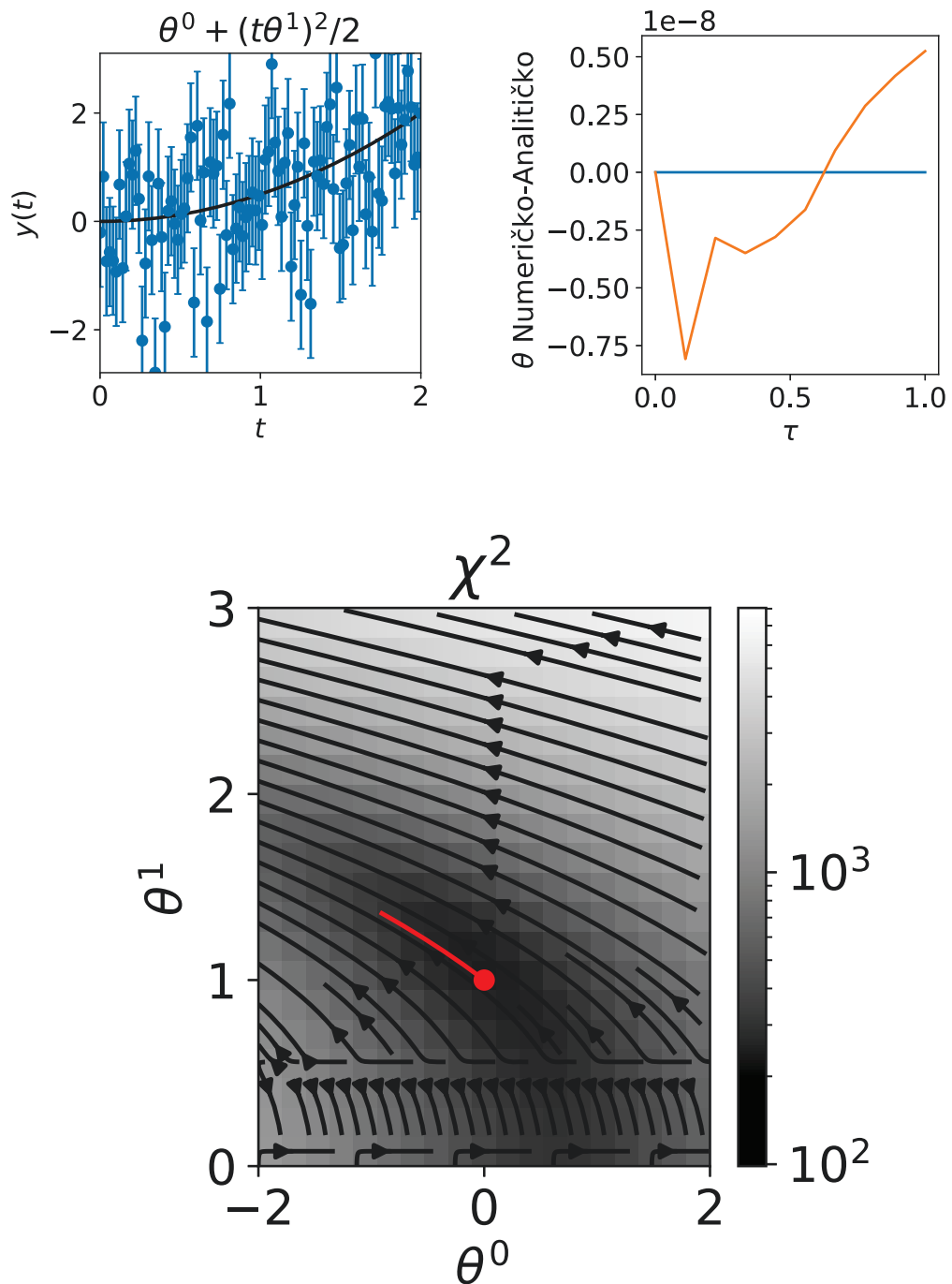
Također, i Riemannov tenzor i frekvencija  $\omega$  su za ovaj model jednaki 0 jer vrijedi

$$\mathbf{H}_{\mu\nu}^b \mathbf{P} \mathbf{H}_{\alpha\beta} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}_{\alpha\beta} \otimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}_{\mu\nu} \mathbf{H}_{11}^b \mathbf{P} \mathbf{H}_{11},$$

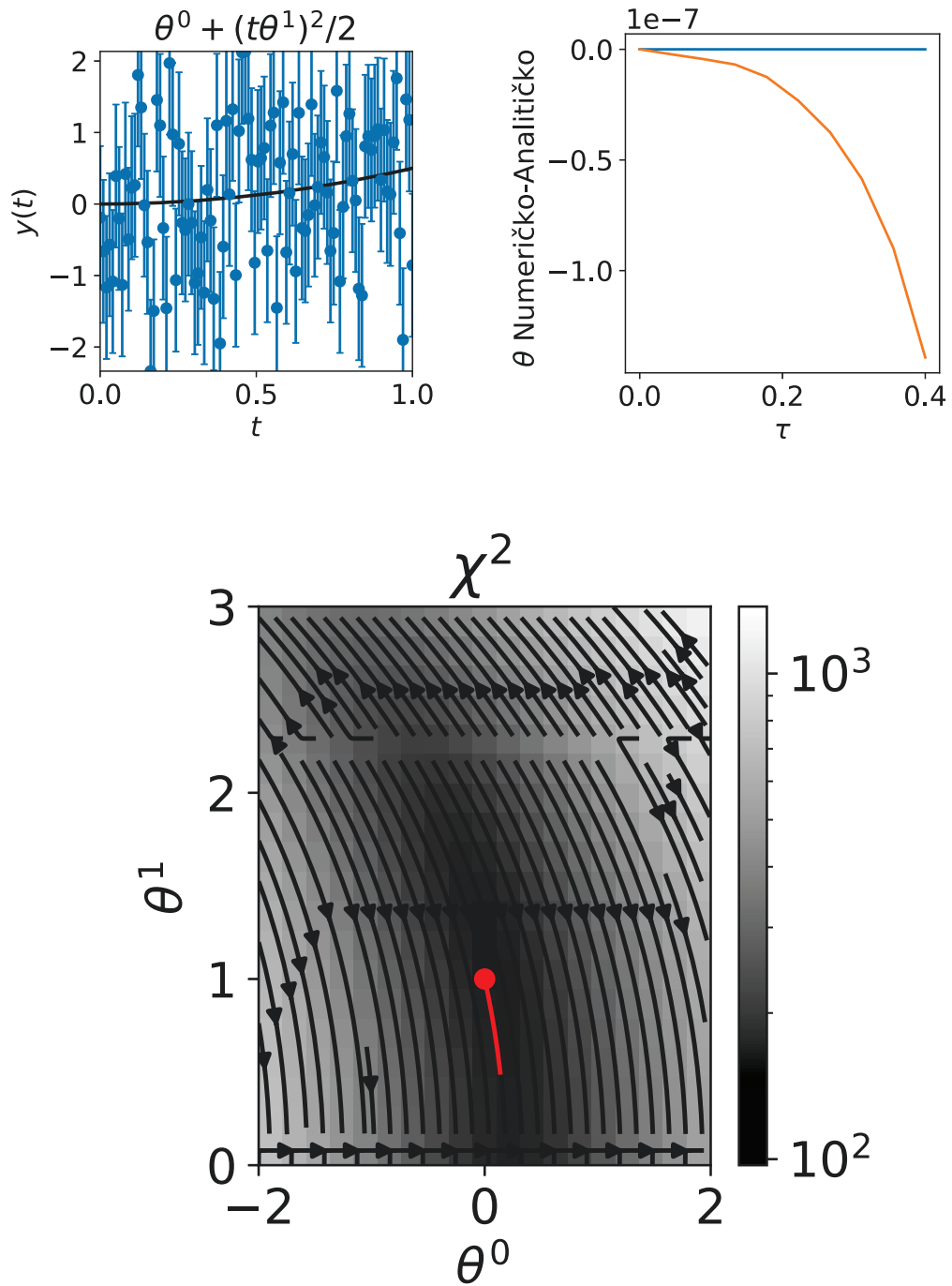
gdje je  $\mathbf{H}_{11}^b \mathbf{P} \mathbf{H}_{11}$  dan s

$$\mathbf{H}_{11}^b \mathbf{P} \mathbf{H}_{11} = (\theta^1)^{n-2}(n-1) \left( \Sigma_{2n} - \frac{\sum_{i=1}^{N_d} \sum_{j=1}^{N_d} \begin{pmatrix} \frac{1}{\sigma_i} & \frac{t_i^n (\theta^1)^{n-1}}{\sigma_i} \end{pmatrix} \begin{pmatrix} \Sigma_{2n} & \frac{-\Sigma_n}{(\theta^1)^{n-1}} \\ \frac{-\Sigma_n}{(\theta^1)^{n-1}} & \frac{\Sigma}{(\theta^1)^{2(n-1)}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma_j} \\ \frac{t_j^n (\theta^1)^{n-1}}{\sigma_j} \end{pmatrix}}{\Sigma_{2n} \Sigma - \Sigma_n^2} \right),$$

što je jednako 0.

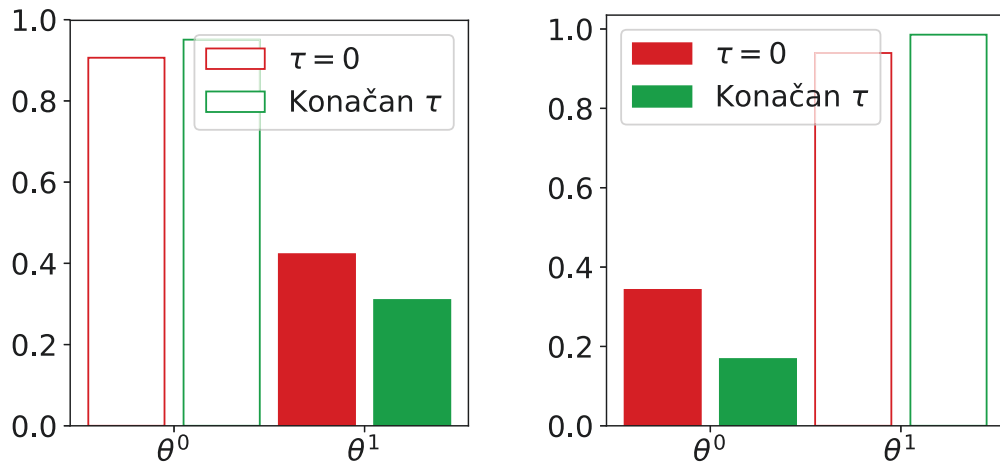


Slika 5.2: Kvadratni model simuliran za podatke  $t \in [0, 2]$ . Oznake iste kao i na slici (5.1). Na gornjem desnom okviru, plavom bojom je označena razlika numeričkog i analitičkog rješenja parametra  $\theta^0$ , a narančastom razlika za  $\theta^1$ .

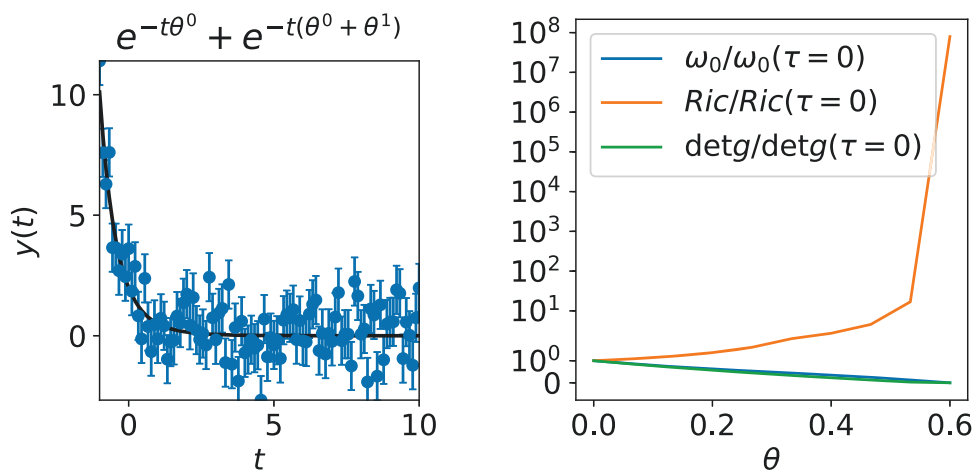


Slika 5.3: Kvadratni model simuliran za podatke  $t \in [0, 1]$ . Oznake iste kao i na slici (5.1). Na gornjem desnom okviru, plavom bojom je označena razlika numeričkog i analitičkog rješenja parametra  $\theta^0$ , a narančastom razlika za  $\theta^1$ .





Slika 5.4: Komponente svojstvenih vektora na početku i kraju geodezika za kvadratni model. Lijevi okvir prikazuje slučaj  $t \in [0, 2]$ , dok desni prikazuje slučaj  $t \in [0, 1]$ .



Slika 5.5: Model 2 eksponencijale (jednadžba 5.7). Model 5.7 je simuliran za podatke  $t \in [0, 10]$ . Oznake iste kao i na slici (5.1).

### 5.3 Modeli konačne zakrivljenosti

U ovom odjeljku promatramo kompliciranije modele, a za koje analitička usporedba nije moguća. Izabrana su dva modela iz [5] koja se razlikuju po iznosu frekvencije  $\omega$ .

Jednostavan model sume eksponencijala

$$f(t, \theta) = \sum_{i=1}^{N_p} e^{-\theta^i t}, \quad (5.5)$$

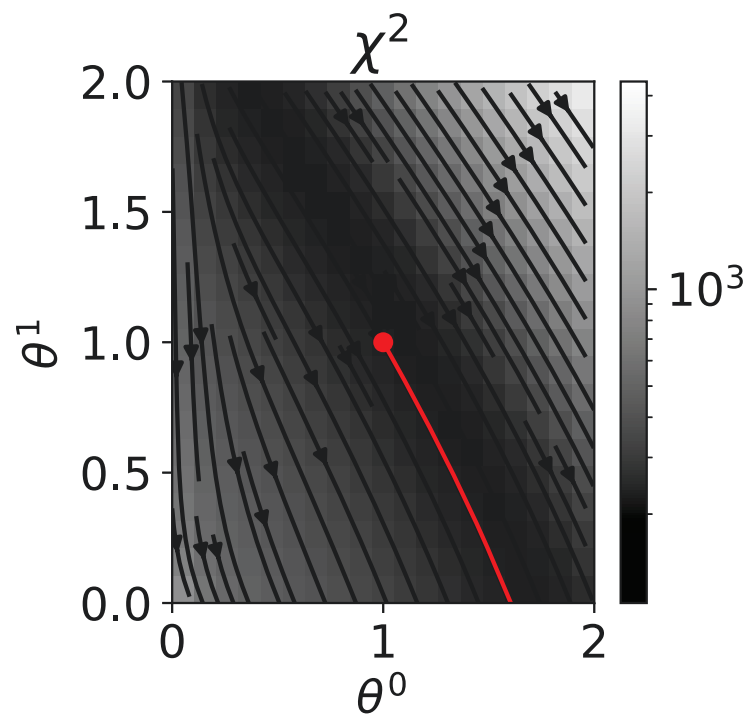
za  $N_p = 2$  i  $N_p = 3$  ima Fisherovu metriku oblika

$$g = \frac{1}{\sigma^2} \begin{pmatrix} \sum t^2 e^{-2\theta^0 t} & \sum t^2 e^{-(\theta^0 + \theta^1)t} \\ \sum t^2 e^{-(\theta^0 + \theta^1)t} & \sum t^2 e^{-2\theta^1 t} \end{pmatrix}, \quad (5.6)$$

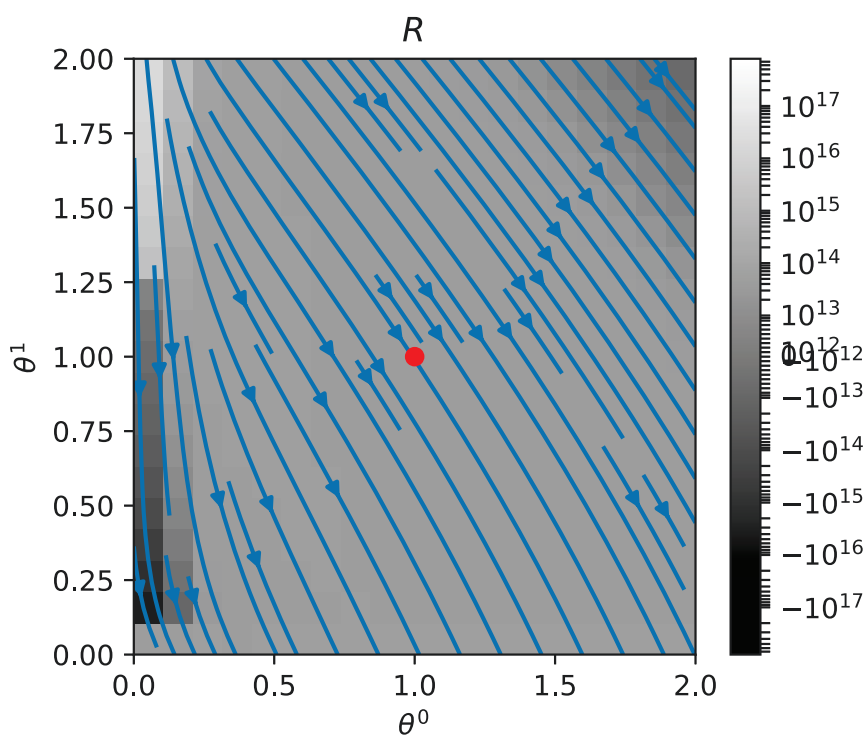
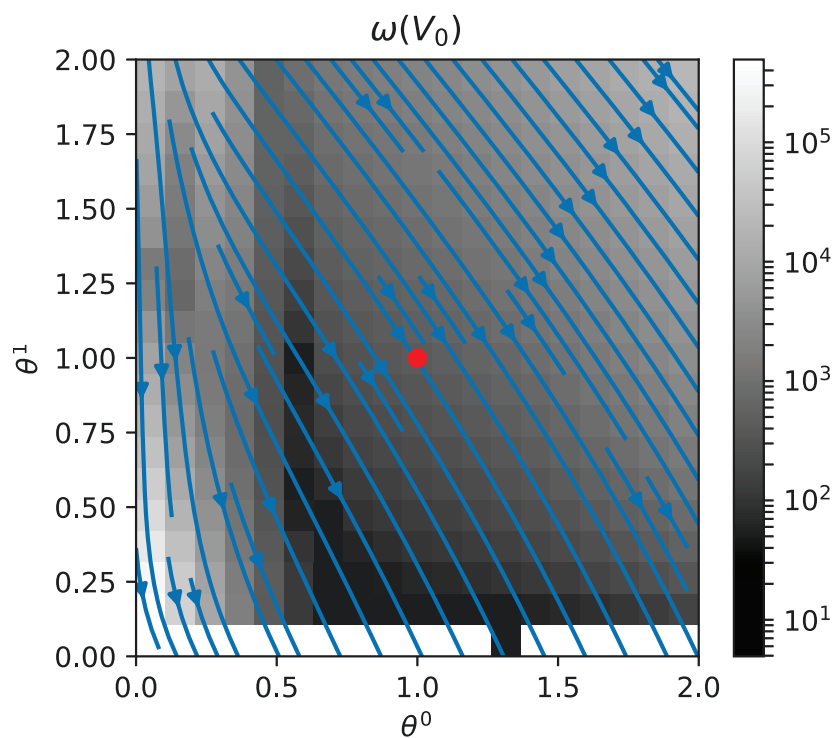
Ova metrika je singularna u slučaju kada je  $\theta^0 = \theta^1$ , odnosno imerzija,  $r : \mathcal{P} \rightarrow \mathbf{r}(\mathcal{P})$ , nema konstantan rang. Model koji bi eliminirao ovakvo ponašanje je

$$f(t, \theta) = \sum_{i=1}^{N_p} e^{-\sum_{j=1}^i \theta^j t}. \quad (5.7)$$

Na slici 5.6 prikazane su vrijednosti  $\chi^2$ , dok su frekvencije  $\omega$  i Riccijevog skalara u okolini točke najbolje prilagodbe prikazane na slici 5.7. Na desnom okviru slike 5.7 prikazane njihove vrijednosti duž geodezijske krivulje. Ovaj model ima frekvenciju  $\omega$  mnogo redova veličine veću od  $2\pi$ , dok Riccijev skalar daleko od točke najbolje prilagodbe mijenja i predznak. Model je ipak jednostavan za opisati jer se njihove vrijednosti ne mijenjaju značajno oko točke najbolje prilagodbe, što prestaje vrijediti na kraju geodezika.



Slika 5.6: Model 2 eksponencijale (jednadžba 5.7) simuliran za podatke  $t \in [0, 2]$ . Vrijednosti  $\chi^2$  za točke u prostoru parametara.



Slika 5.7: Model 2 eksponencijale (jednadžba 5.7) simuliran za podatke  $t \in [0, 2]$ . Gornji red prikazuje frekvencije  $\omega$  za točke prostora parametara, dok donji red prikazuje Riccijev skalar.

## 5.4 Primjer MBAM redukcije na primjeru razlomljenog modela

Dajemo primjer redukcije modela na modelu s više parametara,  $\theta^0, \dots, \theta^6$  opisanom jednadžbom

$$y(t, \theta) = \frac{\theta^0 + t\theta^1 + t^2\theta^2 + t^3\theta^3}{1 + t\theta^4 + t^2\theta^5 + t^3\theta^6}.$$

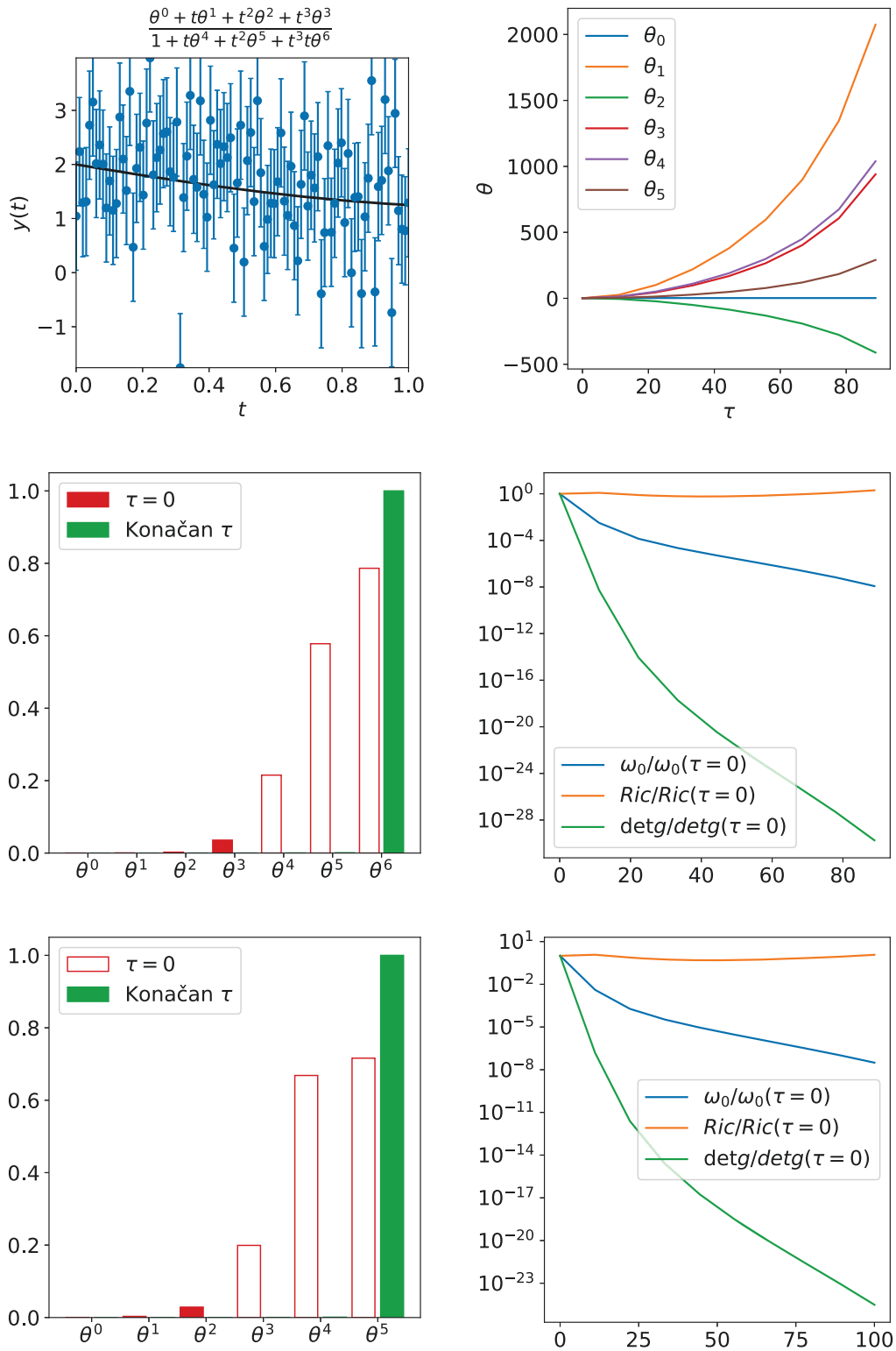
Na slici 5.8 prikazana je redukcija modela. U gornjem redu prikazana su simulirana mjerenja te ovisnost parametara  $\theta$  o 'vremenu' na geodeziku,  $\tau$ . U sljedećem redu prikazan je svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti metrike na početku geodezika,  $\tau = 0$ , te na kraju integracije. Ovakav postupak ukazuje da bi bilo optimalno reducirati model eliminirajući najveću komponentu svojstvenog vektora na kraju geodezika,  $\theta^6$ . Eliminiranje ovog parametra dovodi do modela prikazanog u 3. redu slike 5.8

$$y(t, \theta) = \frac{\theta^0 + t\theta^1 + t^2\theta^2 + t^3\theta^3}{1 + t\theta^4 + t^2\theta^5}.$$

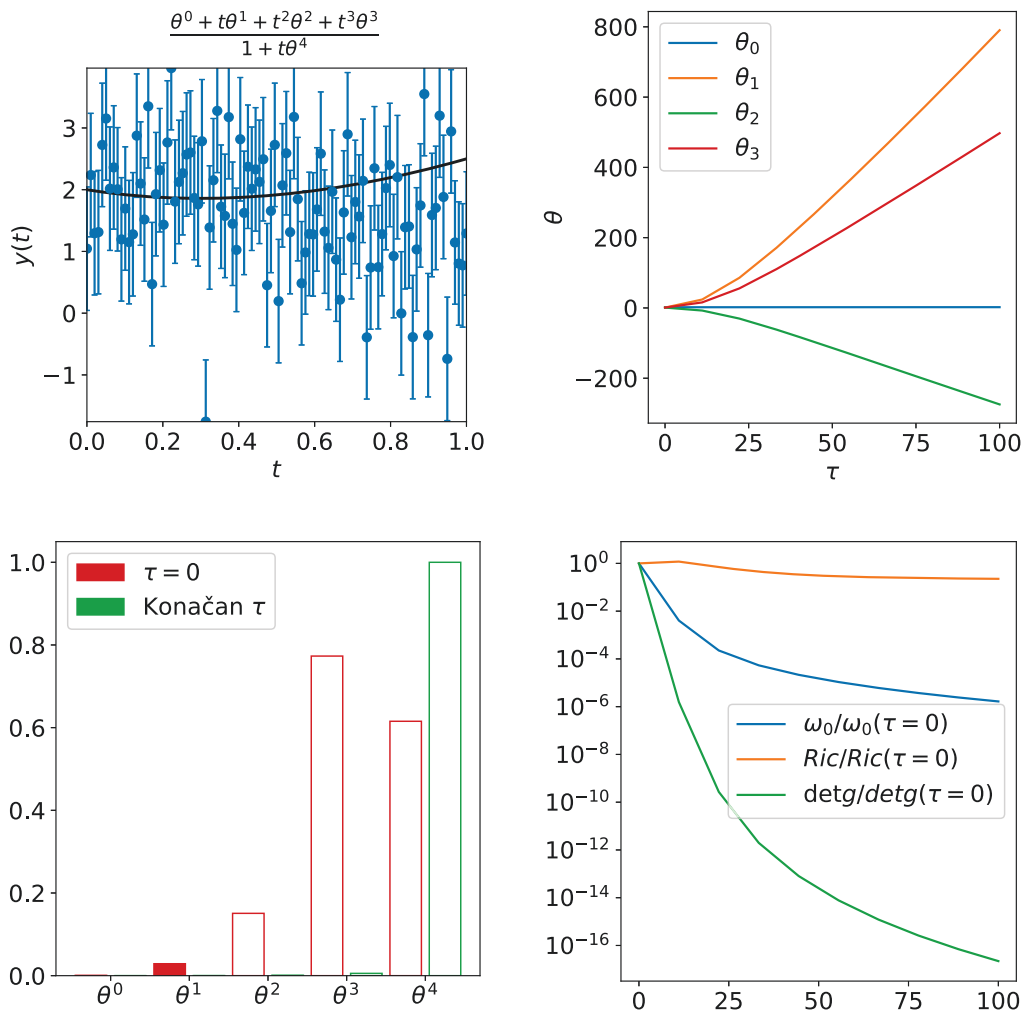
Svojstveni vektor ovako reduciranog modela pak ukazuje na eliminaciju  $\theta^5$ , što dovodi do modela prikazanog na slici 5.9

$$y(t, \theta) = \frac{\theta^0 + t\theta^1 + t^2\theta^2 + t^3\theta^3}{1 + t\theta^4}.$$

Za razlomljeni model nije uočeno neobično ponašanje prilikom integracije, što je korelirano s uniformnim ponašanjem determinante metrike, frekvencije  $\omega$  i Riccijevog skalara za sve izvrijednjenje  $\tau$ .



Slika 5.8: Simulirana mjerenja, parametri modela kao funkcija  $\tau$  te ponašanje svojstvenog vektora metrike koji odgovara njezinoj najmanjoj svojstvenoj vrijednosti za razlomljeni model. Donji red prikazuje svojstveni vektor za jednostruko reducirani model.



Slika 5.9: Simulirana mjerenja, parametri modela kao funkcija  $\tau$  te ponašanje svojstvenog vektora metrike koji odgovara njezinoj najmanjoj svojstvenoj vrijednosti za dvostruko reducirani razlomljeni model.

## 5.5 Primjeri kompleksnih dinamičkih sistema

Kompleksni dinamički sustavi prezentirani u ovom potpoglavlju imaju frekvencije  $\omega \gg 2\pi$  u različitim smjerovima [5]. Štoviše, modeli pokazuju korelaciju između pojavljivanja nestabilnosti numeričkih rješenja i naglih promjena intrinzične i ekstrinzične zakrivljenosti duž geodezika.

### 5.5.1 Troparametarsko njihalo

Prvi primjer dinamičkog sustava je tjerano-gušeno njihalo opisano sustavom jednažbi

$$\dot{\theta} = \omega \quad (5.8)$$

$$\dot{\omega} = -\frac{\omega}{Q} - \sin \theta + A \cos \phi \quad (5.9)$$

$$\dot{\phi} = \omega_D, \quad (5.10)$$

koji ovisi o 3 parametra,  $A$ ,  $Q$  i  $\omega_D$ . Za pregledniji model, koncentriramo se na podmnogostrukost razapetu parametrima  $A \in \mathbb{R}$  i  $Q \in \mathbb{H}^1$  definiranu s  $\omega_D = 0$ . Simulirano je 100 mjerenja za  $t \in [0, 50]$  koristeći normalnu distribuciju sa  $\sigma = 1$  i parametre  $\theta_{bf} = (A, Q) = (1, 1)$  te  $\theta(t = 0) = -2$ ,  $\omega(t = 0) = 0$ . Sustav (5.10) je proširen kako bi se mogla izračunati Fisherova metrika i Christoffelovi simboli.



Ukupni sustav je

$$\dot{\theta} = \omega \quad (5.11)$$

$$\dot{\omega} = -\frac{\omega}{Q} - \sin \theta + A \cos \phi \quad (5.12)$$

$$\dot{\theta}_A = \omega_A \quad (5.13)$$

$$\dot{\omega}_A = -\frac{\omega_A}{Q} - \cos \theta \theta_A + 1 \quad (5.14)$$

$$\dot{\theta}_Q = \omega_Q \quad (5.15)$$

$$\dot{\omega}_Q = -\frac{\omega_Q}{Q} + \frac{\omega}{Q^2} - \cos \theta \theta_Q \quad (5.16)$$

$$\dot{\theta}_{AA} = \omega_{AA} \quad (5.17)$$

$$\dot{\omega}_{AA} = -\frac{\omega_{AA}}{Q} - \cos \theta \theta_{AA} + \sin \theta \theta_A^2 \quad (5.18)$$

$$\dot{\theta}_{AQ} = \omega_{AQ} \quad (5.19)$$

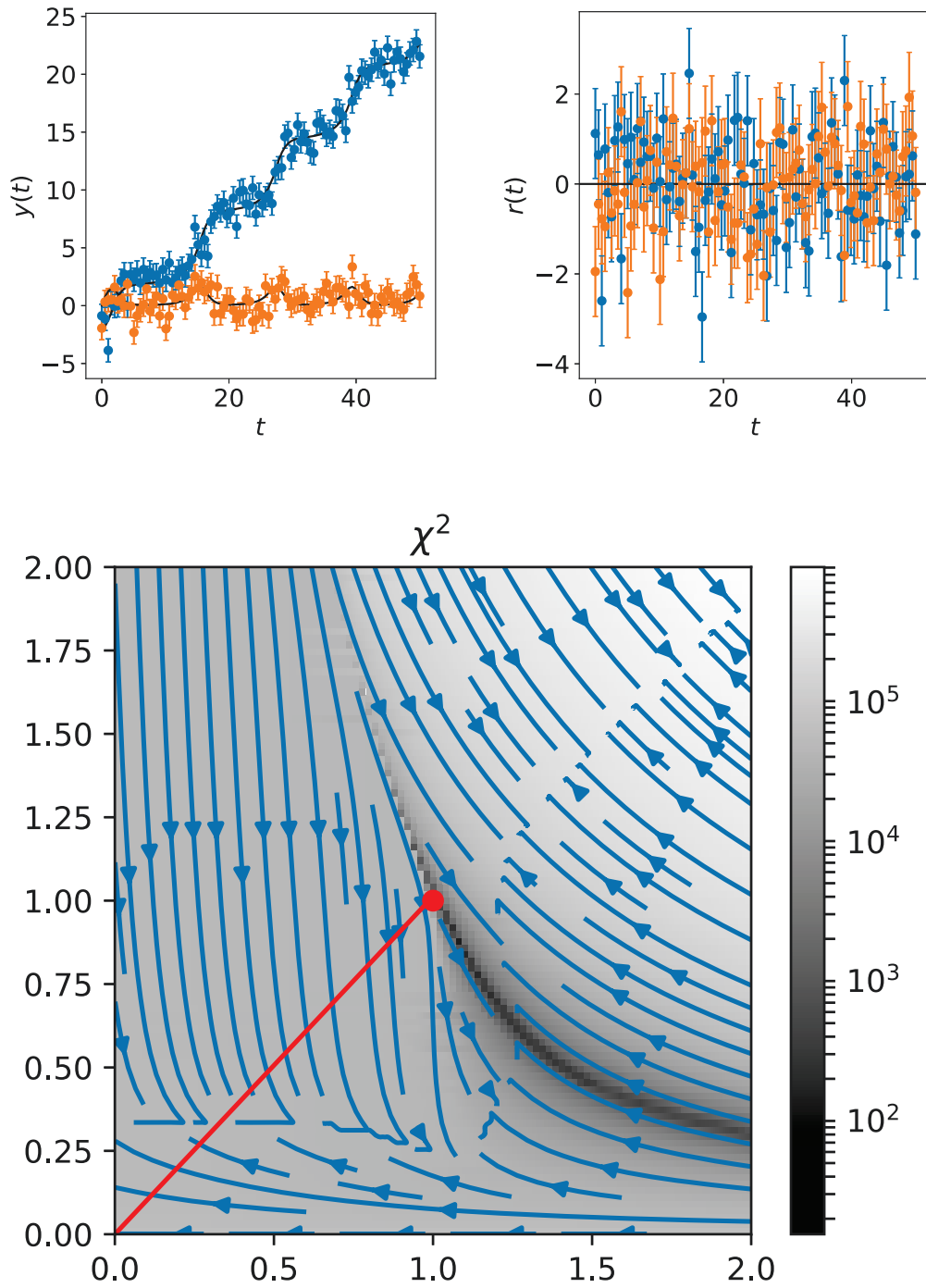
$$\dot{\omega}_{AQ} = -\frac{\omega_{AQ}}{Q} + \frac{\omega_A}{Q^2} + \sin \theta \theta_A \theta_Q - \cos \theta \theta_{AQ} \quad (5.20)$$

$$\dot{\theta}_{QQ} = \omega_{QQ} \quad (5.21)$$

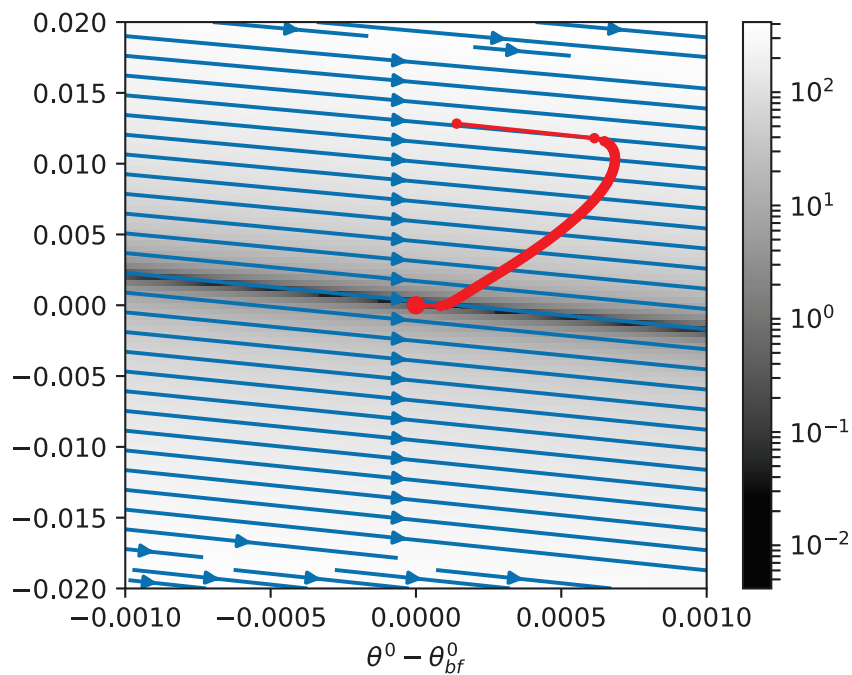
$$\dot{\omega}_{QQ} = -\frac{\omega_{QQ}}{Q} + 2\frac{\omega_Q}{Q^2} - 2\frac{\omega}{Q^3} - \cos \theta \theta_{QQ} + \sin \theta \theta_Q^2. \quad (5.22)$$

Početni uvjet je proširen tako da su sve derivacije  $\theta_i, \omega_i$  za  $i \in \{A, Q, AA, AQ, QQ\}$  jednake nuli u  $t = 0$ . Realizacija simuliranih mjerenja i  $\chi^2$  prikazani su za ovaj model na slici 5.10.

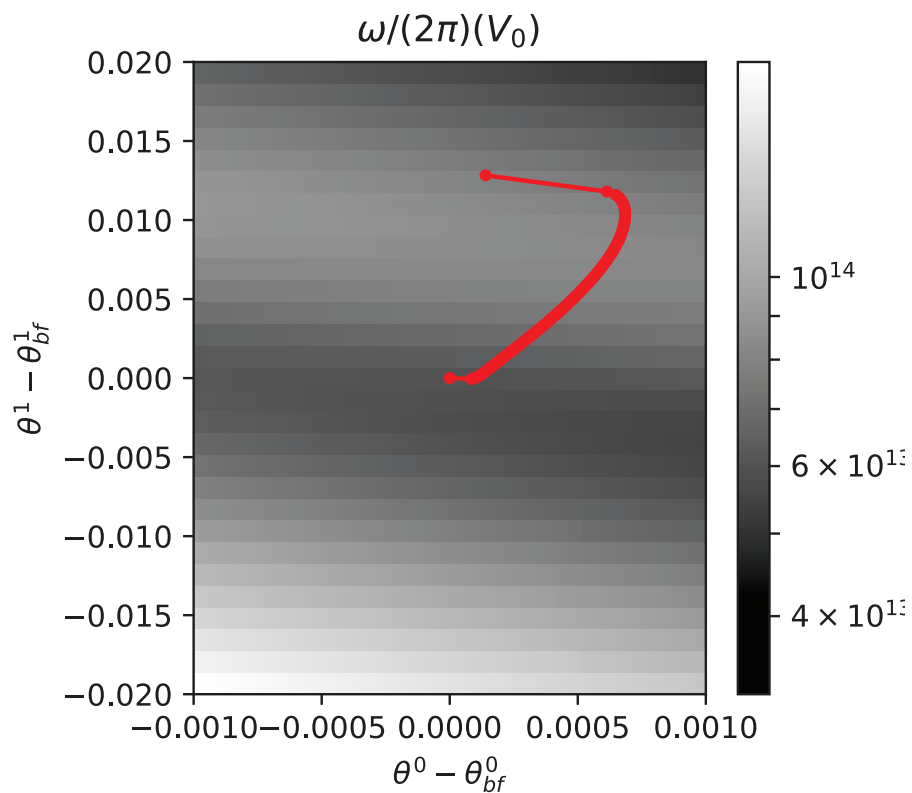
Iako naoko izgleda kao da MBAM daje konačne parametre  $(0, 0)$ , nužno je promotriti bliske točke. Na slici 5.11 prikazano je ponašanje  $\chi^2$  vrijednosti i MBAM geodezika u okolini točke najbolje prilagodbe. Vidimo da su većina točaka geodezika, no vrlo brzo dolazi do divergencije, iako naizgled nema značajnih promjena u  $\chi^2$  vrijednosti. Na slici 5.12 prikazana je frekvencija  $\omega$ , a na slici 5.13 Riccijev skalar za dani model. Iako frekvencija  $\omega$  nema značajne varijacije duž geodezika, Riccijev skalar mijenja predznak iz pozitivnog u negativni, što je vjerojatan uzrok promjene ponašanja geodezika.



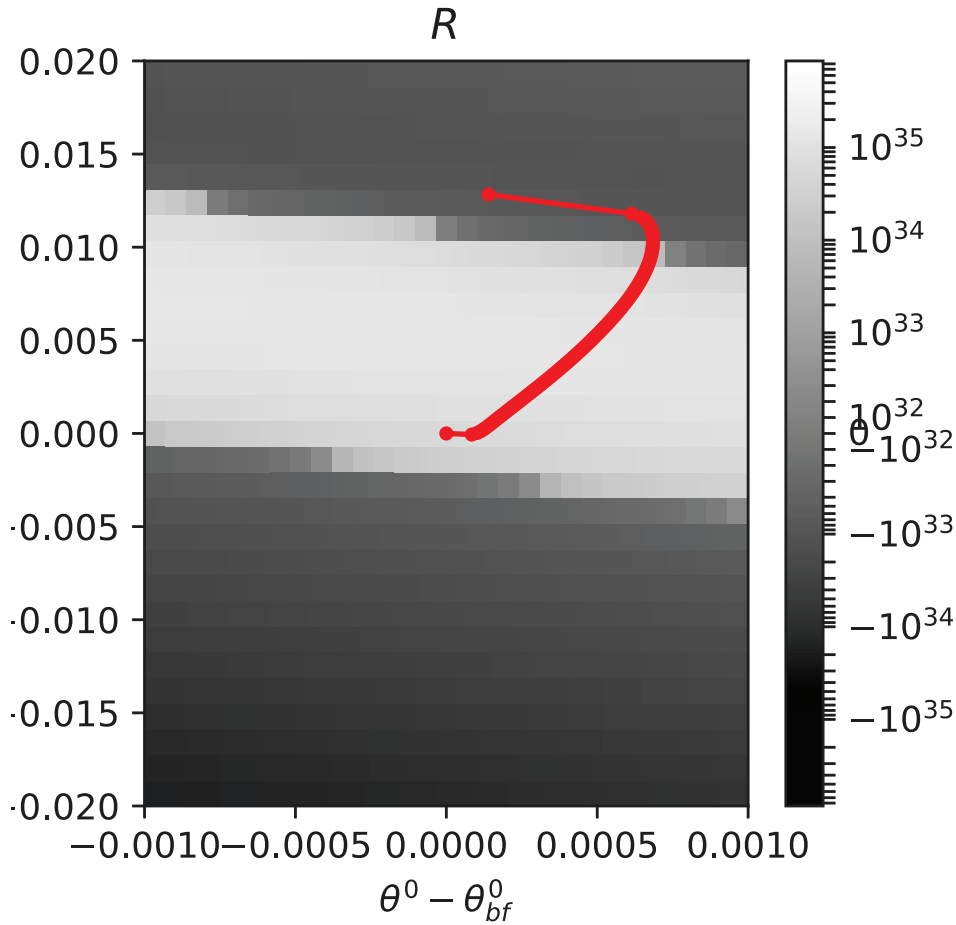
Slika 5.10: Tjerano-gušeno njihalo. Na gornjim okvirima, plavom bojom je označena vremenska ovisnost  $\theta(t)$ , a narančastom  $\omega(t)$ . Označke iste kao i na slici (5.1).



Slika 5.11: Vrijednosti  $\chi^2$  za tjerano-gušeno njihalo za bliske točke oko točke najbolje prilagodbe. Oznake iste kao i na slici (5.1).



Slika 5.12: Vrijednosti  $\omega_0$  za tjerano-gušeno njihalo za bliske točke oko točke najbolje prilagodbe. Oznake iste kao i na slici (5.1).



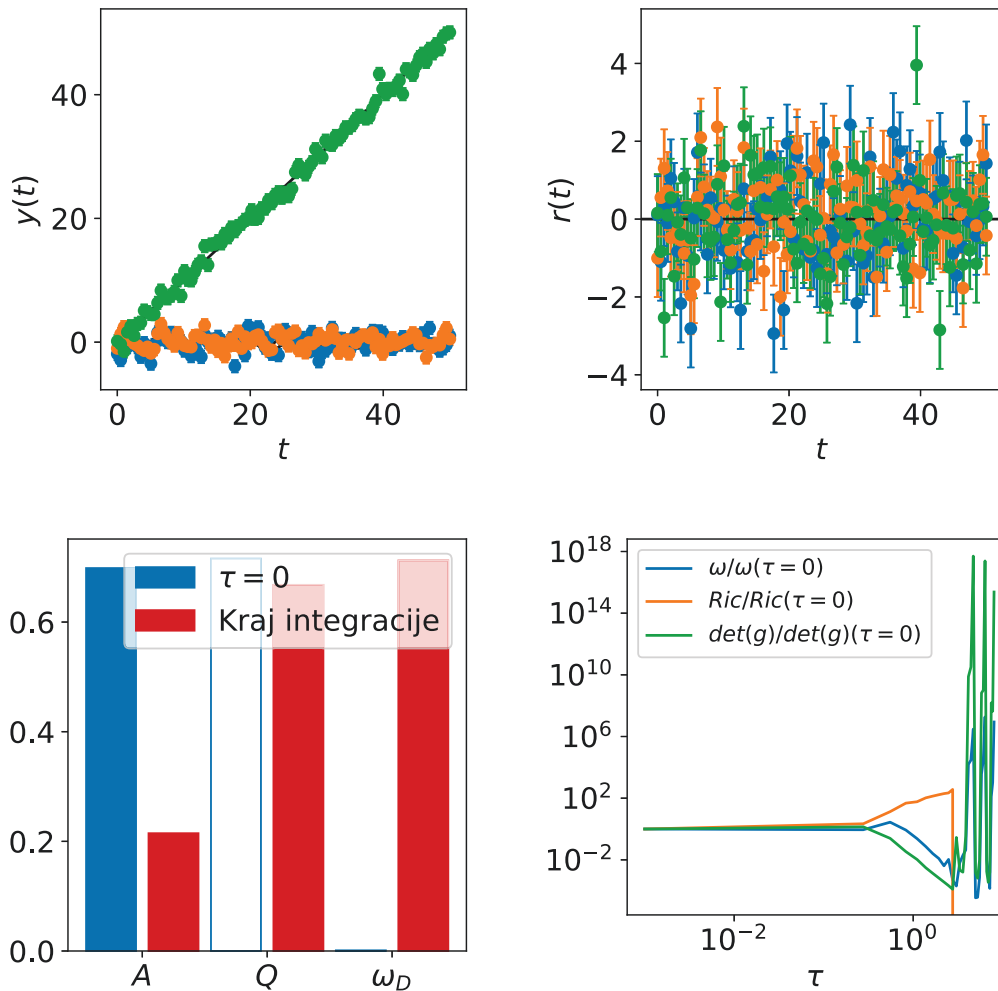
Slika 5.13: Vrijednosti  $\omega_0$  za tjerano-gušeno njihalo za bliske točke oko točke najbolje prilagodbe. Oznake iste kao i na slici (5.1).

Na slici 5.14 prikazana su rješenja duž MBAM geodezika za troparametarsko (3D) tjerano-gušeno njihalo. Na ovom se modelu vidi korelacija između Riccijevog skalara te determinante metrike sa dolaženjem do ruba mnogostrukosti. U nastavku je prikazan sustav jednažbi za koji je provedena integracija kako bi se dobio MBAM geodezik.

$$\begin{aligned}
\frac{d}{dt} \frac{\partial}{\partial A} \theta &= \frac{\partial}{\partial A} \omega \\
\frac{d}{dt} \frac{\partial}{\partial A} \omega &= -A \sin(\varphi) \frac{\partial}{\partial A} \varphi - \cos(\theta) \frac{\partial}{\partial A} \theta + \cos(\varphi) - \frac{1}{Q} \frac{\partial}{\partial A} \omega \\
\frac{d}{dt} \frac{\partial}{\partial A} \varphi &= 0 \\
\frac{d}{dt} \frac{\partial}{\partial Q} \theta &= \frac{\partial}{\partial Q} \omega \\
\frac{d}{dt} \frac{\partial}{\partial Q} \omega &= -A \sin(\varphi) \frac{\partial}{\partial Q} \varphi - \cos(\theta) \frac{\partial}{\partial Q} \theta - \frac{1}{Q} \frac{\partial}{\partial Q} \omega + \frac{1}{Q^2} \omega \\
\frac{d}{dt} \frac{\partial}{\partial Q} \varphi &= 0
\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt} \frac{\partial}{\partial \omega_D} \theta &= \frac{\partial}{\partial \omega_D} \omega \\
\frac{d}{dt} \frac{\partial}{\partial \omega_D} \omega &= -A \sin(\varphi) \frac{\partial}{\partial \omega_D} \varphi - \cos(\theta) \frac{\partial}{\partial \omega_D} \theta - \frac{1}{Q} \frac{\partial}{\partial \omega_D} \omega \\
\frac{d}{dt} \frac{\partial}{\partial \omega_D} \varphi &= 1 \\
\frac{d}{dt} \frac{\partial^2}{\partial A^2} \theta &= \frac{\partial^2}{\partial A^2} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A^2} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial A^2} \varphi - A \cos(\varphi) \left( \frac{\partial}{\partial A} \varphi \right)^2 \\
&\quad + \sin(\theta) \left( \frac{\partial}{\partial A} \theta \right)^2 - 2 \sin(\varphi) \frac{\partial}{\partial A} \varphi - \cos(\theta) \frac{\partial^2}{\partial A^2} \theta - \frac{1}{Q} \frac{\partial^2}{\partial A^2} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A^2} \varphi &= 0 \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial Q} \theta &= \frac{\partial^2}{\partial A \partial Q} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial Q} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial A \partial Q} \varphi - A \cos(\varphi) \frac{\partial}{\partial A} \varphi \frac{\partial}{\partial Q} \varphi + \sin(\theta) \frac{\partial}{\partial A} \theta \frac{\partial}{\partial Q} \theta \\
&\quad - \sin(\varphi) \frac{\partial}{\partial Q} \varphi - \cos(\theta) \frac{\partial^2}{\partial A \partial Q} \theta - \frac{1}{Q} \frac{\partial^2}{\partial A \partial Q} \omega + \frac{1}{Q^2} \frac{\partial}{\partial A} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial Q} \varphi &= 0 \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial \omega_D} \theta &= \frac{\partial^2}{\partial A \partial \omega_D} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial \omega_D} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial A \partial \omega_D} \varphi - A \cos(\varphi) \frac{\partial}{\partial A} \varphi \frac{\partial}{\partial \omega_D} \varphi + \sin(\theta) \frac{\partial}{\partial A} \theta \frac{\partial}{\partial \omega_D} \theta \\
&\quad - \sin(\varphi) \frac{\partial}{\partial \omega_D} \varphi - \cos(\theta) \frac{\partial^2}{\partial A \partial \omega_D} \theta - \frac{1}{Q} \frac{\partial^2}{\partial A \partial \omega_D} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial A \partial \omega_D} \varphi &= 0 \\
\frac{d}{dt} \frac{\partial^2}{\partial Q^2} \theta &= \frac{\partial^2}{\partial Q^2} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial Q^2} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial Q^2} \varphi - A \cos(\varphi) \left( \frac{\partial}{\partial Q} \varphi \right)^2 + \sin(\theta) \left( \frac{\partial}{\partial Q} \theta \right)^2 \\
&\quad - \cos(\theta) \frac{\partial^2}{\partial Q^2} \theta - \frac{1}{Q} \frac{\partial^2}{\partial Q^2} \omega + \frac{2}{Q^2} \frac{\partial}{\partial Q} \omega - \frac{2}{Q^3} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial Q^2} \varphi &= 0 \\
\frac{d}{dt} \frac{\partial^2}{\partial Q \partial \omega_D} \theta &= \frac{\partial^2}{\partial Q \partial \omega_D} \omega \\
\frac{d}{dt} \frac{\partial^2}{\partial Q \partial \omega_D} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial Q \partial \omega_D} \varphi - A \cos(\varphi) \frac{\partial}{\partial Q} \varphi \frac{\partial}{\partial \omega_D} \varphi + \sin(\theta) \frac{\partial}{\partial Q} \theta \frac{\partial}{\partial \omega_D} \theta \\
&\quad - \cos(\theta) \frac{\partial^2}{\partial Q \partial \omega_D} \theta - \frac{1}{Q} \frac{\partial^2}{\partial Q \partial \omega_D} \omega + \frac{1}{Q^2} \frac{\partial}{\partial \omega_D} \omega
\end{aligned}$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial^2}{\partial Q \partial \omega_D} \varphi &= 0 \\ \frac{d}{dt} \frac{\partial^2}{\partial \omega_D^2} \theta &= \frac{\partial^2}{\partial \omega_D^2} \omega \\ \frac{d}{dt} \frac{\partial^2}{\partial \omega_D^2} \omega &= -A \sin(\varphi) \frac{\partial^2}{\partial \omega_D^2} \varphi - A \cos(\varphi) \left( \frac{\partial}{\partial \omega_D} \varphi \right)^2 + \sin(\theta) \left( \frac{\partial}{\partial \omega_D} \theta \right)^2 \\ &\quad - \cos(\theta) \frac{\partial^2}{\partial \omega_D^2} \theta - \frac{1}{Q} \frac{\partial^2}{\partial \omega_D^2} \omega \\ \frac{d}{dt} \frac{\partial^2}{\partial \omega_D^2} \varphi &= 0 \end{aligned}$$



Slika 5.14: Simulirana mjerenja te rješenja duž MBAM geodezika za model tjeranogušenog njihala u 3D: Parametri (srednji red, lijevo), svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti metrike (srednji red, desno) te frekvencija  $\omega$ , Riccijev skalar i determinanta metrike kao funkcija  $\tau$ . Na gornjim okvirima, plavom bojom je označena vremenska ovisnost  $\theta(t)$ , narančastom  $\omega(t)$ , a zelenom  $\phi(t)$ .

### 5.5.2 FitzHugh-Nagumo model

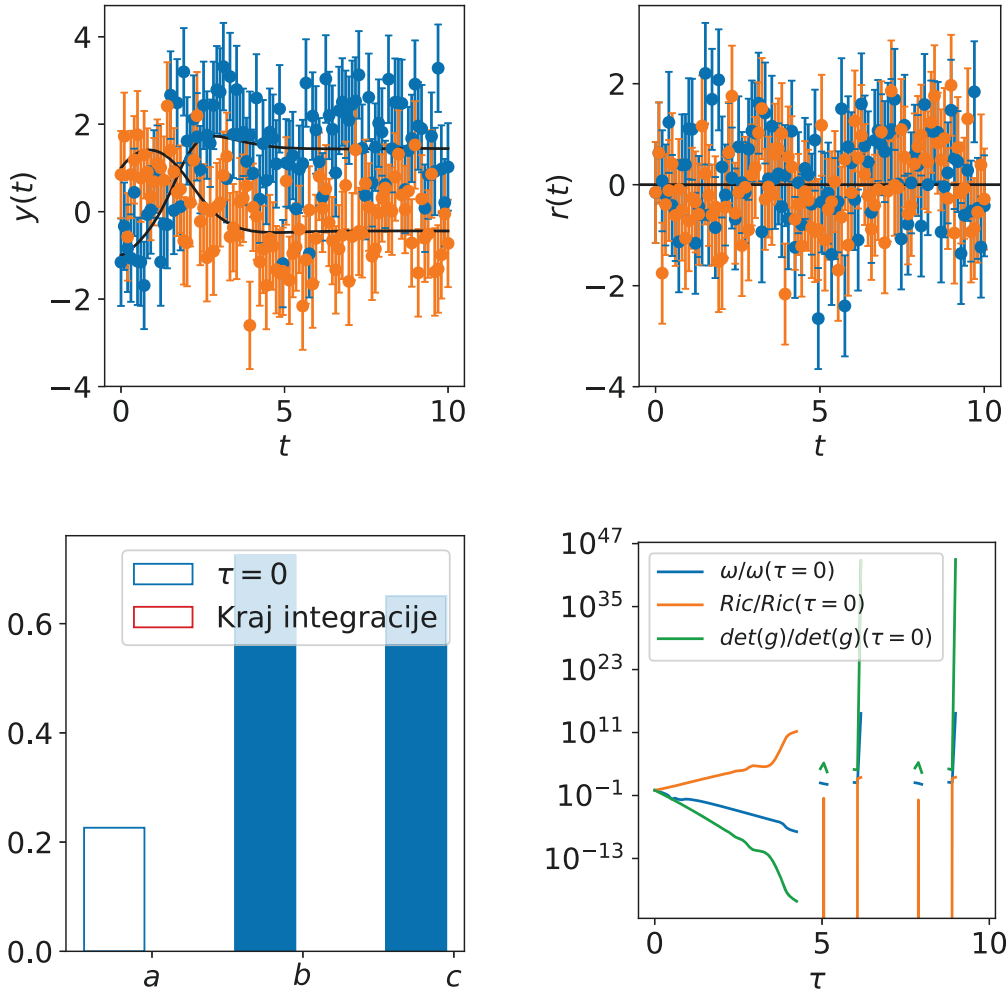
Na slici 5.15 prikazana su rješenja duž MBAM geodezika za FitzHugh-Nagumo model. Na ovom se modelu također vidi korelacija između Riccijevog skalara te determinante metrike sa dolaženjem do ruba mnogostrukosti. U nastavku je prikazan sustav jednažbi za koji je provedena integracija kako bi se dobio MBAM geodezik.

$$\begin{aligned}
\frac{d}{dt} \frac{\partial}{\partial a} V &= c \left( -V^2 \frac{\partial}{\partial a} V + \frac{\partial}{\partial a} R + \frac{\partial}{\partial a} V \right) \\
\frac{d}{dt} \frac{\partial}{\partial a} R &= \frac{1}{c} \left( -b \frac{\partial}{\partial a} R - \frac{\partial}{\partial a} V + 1 \right) \\
\frac{d}{dt} \frac{\partial}{\partial b} V &= c \left( -V^2 \frac{\partial}{\partial b} V + \frac{\partial}{\partial b} R + \frac{\partial}{\partial b} V \right) \\
\frac{d}{dt} \frac{\partial}{\partial b} R &= \frac{1}{c} \left( -b \frac{\partial}{\partial b} R - R - \frac{\partial}{\partial b} V \right) \\
\frac{d}{dt} \frac{\partial}{\partial c} V &= c \left( -V^2 \frac{\partial}{\partial c} V + \frac{\partial}{\partial c} R + \frac{\partial}{\partial c} V \right) + R - \frac{1}{3} V^3 + V \\
\frac{d}{dt} \frac{\partial}{\partial c} R &= \frac{1}{c} \left( -b \frac{\partial}{\partial c} R - \frac{\partial}{\partial c} V \right) - \frac{1}{c^2} (a - bR - V) \\
\frac{d}{dt} \frac{\partial^2}{\partial a^2} V &= -c \left( V^2 \frac{\partial^2}{\partial a^2} V + 2V \left( \frac{\partial}{\partial a} V \right)^2 - \frac{\partial^2}{\partial a^2} R - \frac{\partial^2}{\partial a^2} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial a^2} R &= -\frac{1}{c} \left( b \frac{\partial^2}{\partial a^2} R + \frac{\partial^2}{\partial a^2} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial a \partial b} V &= -c \left( V^2 \frac{\partial^2}{\partial a \partial b} V + 2V \frac{\partial}{\partial a} V \frac{\partial}{\partial b} V - \frac{\partial^2}{\partial a \partial b} R - \frac{\partial^2}{\partial a \partial b} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial a \partial b} R &= -\frac{1}{c} \left( b \frac{\partial^2}{\partial a \partial b} R + \frac{\partial}{\partial a} R + \frac{\partial^2}{\partial a \partial b} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial a \partial c} V &= -c \left( V^2 \frac{\partial^2}{\partial a \partial c} V + 2V \frac{\partial}{\partial a} V \frac{\partial}{\partial c} V - \frac{\partial^2}{\partial a \partial c} R - \frac{\partial^2}{\partial a \partial c} V \right) - V^2 \frac{\partial}{\partial a} V + \frac{\partial}{\partial a} R + \frac{\partial}{\partial a} V \\
\frac{d}{dt} \frac{\partial^2}{\partial a \partial c} R &= \frac{1}{c} \left( -b \frac{\partial^2}{\partial a \partial c} R - \frac{\partial^2}{\partial a \partial c} V + \frac{1}{c} \left( b \frac{\partial}{\partial a} R + \frac{\partial}{\partial a} V - 1 \right) \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial b^2} V &= -c \left( V^2 \frac{\partial^2}{\partial b^2} V + 2V \left( \frac{\partial}{\partial b} V \right)^2 - \frac{\partial^2}{\partial b^2} R - \frac{\partial^2}{\partial b^2} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial b^2} R &= -\frac{1}{c} \left( b \frac{\partial^2}{\partial b^2} R + 2 \frac{\partial}{\partial b} R + \frac{\partial^2}{\partial b^2} V \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial b \partial c} V &= -c \left( V^2 \frac{\partial^2}{\partial b \partial c} V + 2V \frac{\partial}{\partial b} V \frac{\partial}{\partial c} V - \frac{\partial^2}{\partial b \partial c} R - \frac{\partial^2}{\partial b \partial c} V \right) - V^2 \frac{\partial}{\partial b} V + \frac{\partial}{\partial b} R + \frac{\partial}{\partial b} V
\end{aligned}$$

$$\frac{d}{dt} \frac{\partial^2}{\partial b \partial c} R = \frac{1}{c} \left( -b \frac{\partial^2}{\partial b \partial c} R - \frac{\partial}{\partial c} R - \frac{\partial^2}{\partial b \partial c} V + \frac{1}{c} \left( b \frac{\partial}{\partial b} R + R + \frac{\partial}{\partial b} V \right) \right)$$

$$\frac{d}{dt} \frac{\partial^2}{\partial c^2} V = -c \left( V^2 \frac{\partial^2}{\partial c^2} V + 2V \left( \frac{\partial}{\partial c} V \right)^2 - \frac{\partial^2}{\partial c^2} R - \frac{\partial^2}{\partial c^2} V \right) - 2V^2 \frac{\partial}{\partial c} V + 2 \frac{\partial}{\partial c} R + 2 \frac{\partial}{\partial c} V$$

$$\frac{d}{dt} \frac{\partial^2}{\partial c^2} R = \frac{1}{c} \left( -b \frac{\partial^2}{\partial c^2} R - \frac{\partial^2}{\partial c^2} V + \frac{2}{c} \left( b \frac{\partial}{\partial c} R + \frac{\partial}{\partial c} V \right) - \frac{2}{c^2} (-a + bR + V) \right)$$



Slika 5.15: Simulirana mjerenja te rješenja duž MBAM geodezika za FitzHugh-Nagumo model: Parametri (srednji red, lijevo), svojstveni vektor koji odgovara najmanjoj svojstvenoj vrijednosti metrike (srednji red, desno) te frekvencija  $\omega$ , Riccijev skalar i determinanta metrike kao funkcija  $\tau$ . Na gornjim okvirima, plavom bojom je označena vremenska ovisnost  $V(t)$ , a narančastom  $R(t)$ .

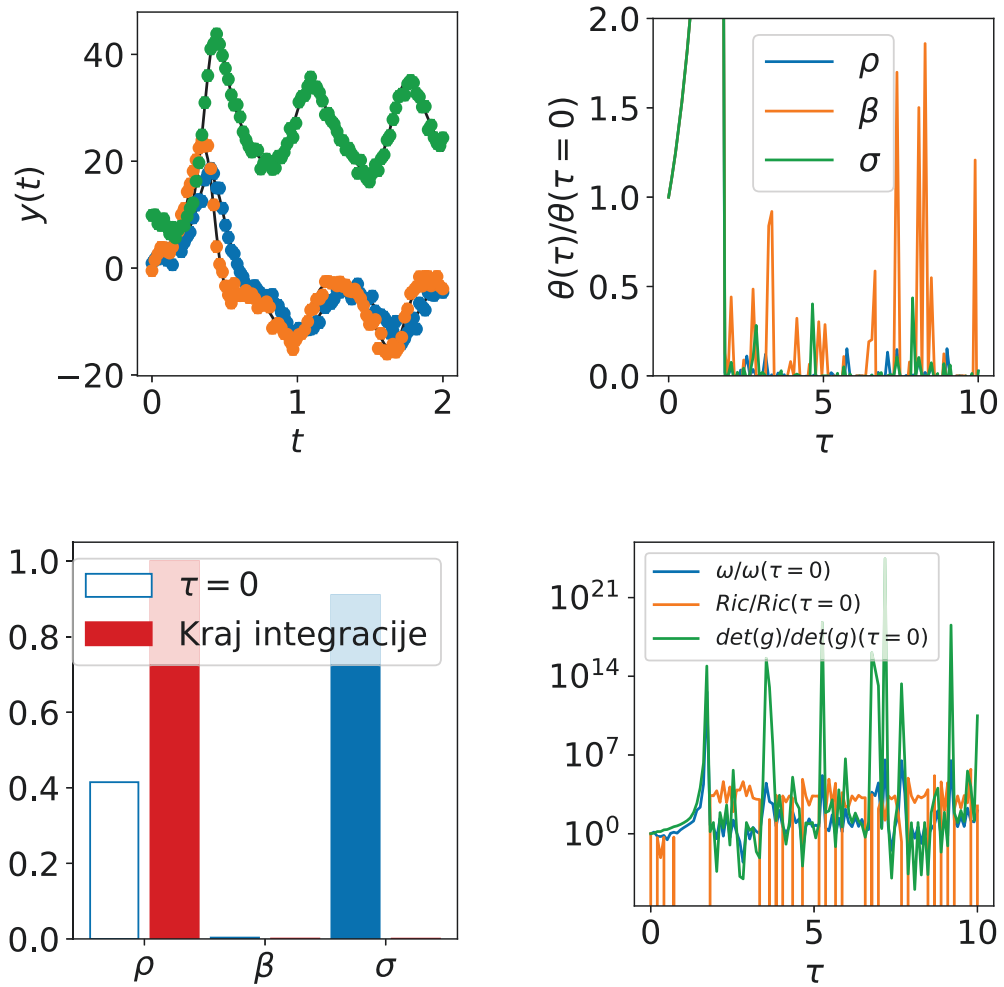


### 5.5.3 Lorenzov model

Na slici 5.16 prikazana su rješenja duž MBAM geodezika za Lorenzov model. Na ovom se modelu također vidi korelacija između Riccijevog skalara te determinante metrike sa dolaženjem do ruba mnogostrukosti. U nastavku je prikazan sustav jednažbi za koji je provedena integracija kako bi se dobio MBAM geodezik.

$$\begin{aligned}
\frac{d}{dt} \frac{\partial}{\partial \rho} x &= \sigma \left( -\frac{\partial}{\partial \rho} x + \frac{\partial}{\partial \rho} y \right) \\
\frac{d}{dt} \frac{\partial}{\partial \rho} y &= (\rho - z) \frac{\partial}{\partial \rho} x + \left( -\frac{\partial}{\partial \rho} z + 1 \right) x - \frac{\partial}{\partial \rho} y \\
\frac{d}{dt} \frac{\partial}{\partial \rho} z &= -\beta \frac{\partial}{\partial \rho} z + x \frac{\partial}{\partial \rho} y + y \frac{\partial}{\partial \rho} x \\
\frac{d}{dt} \frac{\partial}{\partial \beta} x &= \sigma \left( -\frac{\partial}{\partial \beta} x + \frac{\partial}{\partial \beta} y \right) \\
\frac{d}{dt} \frac{\partial}{\partial \beta} y &= (\rho - z) \frac{\partial}{\partial \beta} x - x \frac{\partial}{\partial \beta} z - \frac{\partial}{\partial \beta} y \\
\frac{d}{dt} \frac{\partial}{\partial \beta} z &= -\beta \frac{\partial}{\partial \beta} z + x \frac{\partial}{\partial \beta} y + y \frac{\partial}{\partial \beta} x - z \\
\frac{d}{dt} \frac{\partial}{\partial \sigma} x &= \sigma \left( -\frac{\partial}{\partial \sigma} x + \frac{\partial}{\partial \sigma} y \right) - x + y \\
\frac{d}{dt} \frac{\partial}{\partial \sigma} y &= (\rho - z) \frac{\partial}{\partial \sigma} x - x \frac{\partial}{\partial \sigma} z - \frac{\partial}{\partial \sigma} y \\
\frac{d}{dt} \frac{\partial}{\partial \sigma} z &= -\beta \frac{\partial}{\partial \sigma} z + x \frac{\partial}{\partial \sigma} y + y \frac{\partial}{\partial \sigma} x \\
\frac{d}{dt} \frac{\partial^2}{\partial \rho^2} x &= -\sigma \left( \frac{\partial^2}{\partial \rho^2} x - \frac{\partial^2}{\partial \rho^2} y \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial \rho^2} y &= (\rho - z) \frac{\partial^2}{\partial \rho^2} x - 2 \left( \frac{\partial}{\partial \rho} z - 1 \right) \frac{\partial}{\partial \rho} x - x \frac{\partial^2}{\partial \rho^2} z - \frac{\partial^2}{\partial \rho^2} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \rho^2} z &= -\beta \frac{\partial^2}{\partial \rho^2} z + x \frac{\partial^2}{\partial \rho^2} y + y \frac{\partial^2}{\partial \rho^2} x + 2 \frac{\partial}{\partial \rho} x \frac{\partial}{\partial \rho} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \rho} x &= -\sigma \left( \frac{\partial^2}{\partial \beta \partial \rho} x - \frac{\partial^2}{\partial \beta \partial \rho} y \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \rho} y &= (\rho - z) \frac{\partial^2}{\partial \beta \partial \rho} x - \left( \frac{\partial}{\partial \rho} z - 1 \right) \frac{\partial}{\partial \beta} x - x \frac{\partial^2}{\partial \beta \partial \rho} z - \frac{\partial}{\partial \rho} x \frac{\partial}{\partial \beta} z - \frac{\partial^2}{\partial \beta \partial \rho} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \rho} z &= -\beta \frac{\partial^2}{\partial \beta \partial \rho} z + x \frac{\partial^2}{\partial \beta \partial \rho} y + y \frac{\partial^2}{\partial \beta \partial \rho} x + \frac{\partial}{\partial \beta} x \frac{\partial}{\partial \rho} y + \frac{\partial}{\partial \rho} x \frac{\partial}{\partial \beta} y - \frac{\partial}{\partial \rho} z \\
\frac{d}{dt} \frac{\partial^2}{\partial \rho \partial \sigma} x &= -\sigma \left( \frac{\partial^2}{\partial \rho \partial \sigma} x - \frac{\partial^2}{\partial \rho \partial \sigma} y \right) - \frac{\partial}{\partial \rho} x + \frac{\partial}{\partial \rho} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \rho \partial \sigma} y &= (\rho - z) \frac{\partial^2}{\partial \rho \partial \sigma} x - \left( \frac{\partial}{\partial \rho} z - 1 \right) \frac{\partial}{\partial \sigma} x - x \frac{\partial^2}{\partial \rho \partial \sigma} z - \frac{\partial}{\partial \rho} x \frac{\partial}{\partial \sigma} z - \frac{\partial^2}{\partial \rho \partial \sigma} y
\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt} \frac{\partial^2}{\partial \rho \partial \sigma} z &= -\beta \frac{\partial^2}{\partial \rho \partial \sigma} z + x \frac{\partial^2}{\partial \rho \partial \sigma} y + y \frac{\partial^2}{\partial \rho \partial \sigma} x + \frac{\partial}{\partial \rho} x \frac{\partial}{\partial \sigma} y + \frac{\partial}{\partial \sigma} x \frac{\partial}{\partial \rho} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta^2} x &= -\sigma \left( \frac{\partial^2}{\partial \beta^2} x - \frac{\partial^2}{\partial \beta^2} y \right) \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta^2} y &= (\rho - z) \frac{\partial^2}{\partial \beta^2} x - x \frac{\partial^2}{\partial \beta^2} z - 2 \frac{\partial}{\partial \beta} x \frac{\partial}{\partial \beta} z - \frac{\partial^2}{\partial \beta^2} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta^2} z &= -\beta \frac{\partial^2}{\partial \beta^2} z + x \frac{\partial^2}{\partial \beta^2} y + y \frac{\partial^2}{\partial \beta^2} x + 2 \frac{\partial}{\partial \beta} x \frac{\partial}{\partial \beta} y - 2 \frac{\partial}{\partial \beta} z \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \sigma} x &= -\sigma \left( \frac{\partial^2}{\partial \beta \partial \sigma} x - \frac{\partial^2}{\partial \beta \partial \sigma} y \right) - \frac{\partial}{\partial \beta} x + \frac{\partial}{\partial \beta} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \sigma} y &= (\rho - z) \frac{\partial^2}{\partial \beta \partial \sigma} x - x \frac{\partial^2}{\partial \beta \partial \sigma} z - \frac{\partial}{\partial \beta} x \frac{\partial}{\partial \sigma} z - \frac{\partial}{\partial \sigma} x \frac{\partial}{\partial \beta} z - \frac{\partial^2}{\partial \beta \partial \sigma} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \beta \partial \sigma} z &= -\beta \frac{\partial^2}{\partial \beta \partial \sigma} z + x \frac{\partial^2}{\partial \beta \partial \sigma} y + y \frac{\partial^2}{\partial \beta \partial \sigma} x + \frac{\partial}{\partial \beta} x \frac{\partial}{\partial \sigma} y + \frac{\partial}{\partial \sigma} x \frac{\partial}{\partial \beta} y - \frac{\partial}{\partial \sigma} z \\
\frac{d}{dt} \frac{\partial^2}{\partial \sigma^2} x &= -\sigma \left( \frac{\partial^2}{\partial \sigma^2} x - \frac{\partial^2}{\partial \sigma^2} y \right) - 2 \frac{\partial}{\partial \sigma} x + 2 \frac{\partial}{\partial \sigma} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \sigma^2} y &= (\rho - z) \frac{\partial^2}{\partial \sigma^2} x - x \frac{\partial^2}{\partial \sigma^2} z - 2 \frac{\partial}{\partial \sigma} x \frac{\partial}{\partial \sigma} z - \frac{\partial^2}{\partial \sigma^2} y \\
\frac{d}{dt} \frac{\partial^2}{\partial \sigma^2} z &= -\beta \frac{\partial^2}{\partial \sigma^2} z + x \frac{\partial^2}{\partial \sigma^2} y + y \frac{\partial^2}{\partial \sigma^2} x + 2 \frac{\partial}{\partial \sigma} x \frac{\partial}{\partial \sigma} y
\end{aligned}$$



Slika 5.16: Simulirana mjerenja te rješenja duž MBAM geodezika za Lorenzov model. Na gornjim okvirima, plavom bojom je označena vremenska ovisnost  $x(t)$ , narandžastom parametar  $y(t)$ , a zelenom  $z(t)$ .

## 6 Primjena MBAM metode nuklearne modele

Relativistički Lagrangian temelji se na osnovnim bilinearnim strujama Diracovog spinora  $\psi$  kojim opisujemo nukleone u atomskoj jezgri

$$\bar{\psi}\mathcal{O}_\tau\Gamma\psi, \mathcal{O}_\tau \in 1, \tau_i, \Gamma \in \{1, \gamma_\mu, \gamma_5, \gamma_5\gamma_\mu, \sigma_{\mu\nu}\}. \quad (6.1)$$

$\tau_i$  predstavljaju izospinske Paulijeve matrice, a  $\Gamma$  općenito označava Diracove matrice. Članovi međudjelovanja Lagrangiana su produkti tih bilinearnih kombinacija. U principu, općeniti efektivni Lagrangian možemo zapisati kao razvoj po strujama i njihovim derivacijama. Za ovaj model razmatramo sljedeće četiri fermionske komponente međudjelovanja: izoskalar-skalar:  $(\bar{\psi}\psi)^2$  izoskalar-vektor:  $(\bar{\psi}\gamma_\mu\psi)(\bar{\psi}\gamma^\mu\psi)$  izovektor-skalar:  $(\bar{\psi}\vec{\tau}\psi) \cdot (\bar{\psi}\vec{\tau}\psi)$  izovektor-vektor:  $(\bar{\psi}\vec{\tau}\gamma_\mu\psi) \cdot (\bar{\psi}\vec{\tau}\gamma^\mu\psi)$ . Vektori u izospinskom prostoru su označeni strelicama. Bitno je naglasiti da svakom od elemenata iz skupa  $\bar{\psi}\mathcal{O}_\tau\Gamma\psi$  pridjeljujemo funkciju vrha ( $D_S, D_V, G_S, G_V, G_{TS}$  i  $G_{TV}$ ) kojom ih vežemo u jedan Lagrangian, pri čemu funkcije vrha ovise o barionskoj gustoći  $\hat{\rho} = \bar{\psi}\psi$ . Model je definiran gustoćom Lagrangiana [10]

$$\mathcal{L} = \mathcal{L}_{free} + \mathcal{L}_{4f} + \mathcal{L}_{der} + \mathcal{L}_{em} \quad (6.2)$$

sa sljedećim komponentama

$$\mathcal{L}_{free} = \bar{\psi}(i\gamma_\mu\partial^\mu - M)\psi, \quad (6.3)$$

$$\begin{aligned} \mathcal{L}_{4f} = & -\frac{1}{2}G_S(\hat{\rho})(\bar{\psi}\psi)(\bar{\psi}\psi) - \frac{1}{2}G_V(\hat{\rho})(\bar{\psi}\gamma_\mu\psi)(\bar{\psi}\gamma^\mu\psi) \\ & -\frac{1}{2}G_{TS}(\hat{\rho})(\bar{\psi}\vec{\tau}\psi) \cdot (\bar{\psi}\vec{\tau}\psi) - \frac{1}{2}G_{TV}(\hat{\rho})(\bar{\psi}\vec{\tau}\gamma_\mu\psi) \cdot (\bar{\psi}\vec{\tau}\gamma^\mu\psi), \end{aligned} \quad (6.4)$$

$$\mathcal{L}_{der} = -\frac{1}{2}D_S(\hat{\rho})(\partial_\nu\bar{\psi}\psi)(\partial^\nu\bar{\psi}\psi) - \frac{1}{2}D_V(\hat{\rho})(\partial_\mu\bar{\psi}\gamma_\mu\psi)(\partial^\mu\bar{\psi}\gamma_\mu\psi) + O(\partial^2), \quad (6.5)$$

$$\mathcal{L}_{em} = eA^\mu\bar{\psi}\frac{1+\tau_3}{2}\gamma_\mu\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu}. \quad (6.6)$$

## 6.1 Komponente Diracove jednadžbe

Jednadžbu (6.2) možemo varirati po  $\bar{\psi}$ , što nam daje Diracovu jednadžbu

$$[\gamma_\mu(i\partial^\mu - V^\mu) - (M + S)]\psi = 0, \quad (6.7)$$

gdje su

$$S = \Sigma_S + \vec{\tau} \cdot \vec{\Sigma}_{TS} + \Sigma_{rS}, \quad (6.8)$$

$$V^\mu = \Sigma^\mu + \vec{\tau} \cdot \vec{\Sigma}_S + \Sigma_{rS}, \quad (6.9)$$

pri čemu je

$$\Sigma^\mu = G_V(\bar{\psi}\gamma^\mu\psi) - eA^\mu\frac{1 + \tau_3}{2}, \quad (6.10)$$

$$\vec{\Sigma}_T^\mu = G_{TV}(\bar{\psi}\vec{\tau}\gamma^\mu\psi), \quad (6.11)$$

$$\Sigma_S = G_S(\bar{\psi}\psi) - D_S\Box(\bar{\psi}\psi), \quad (6.12)$$

$$\vec{\Sigma}_{TS} = G_{TS}(\bar{\psi}\vec{\tau}\psi), \quad (6.13)$$

$$\Sigma_{rS} = -\frac{\partial D_S}{\partial \hat{\rho}}(\partial_\nu j^\mu)u_\mu(\partial^\nu(\bar{\psi}\psi)), \quad (6.14)$$

$$\Sigma_r^\mu = \frac{u^\mu}{2}\left(\frac{\partial G_S}{\partial \hat{\rho}}(\bar{\psi}\psi)(\bar{\psi}\psi) + \left(\frac{\partial G_{TS}}{\partial \hat{\rho}}(\bar{\psi}\vec{\tau}\psi) \cdot (\bar{\psi}\vec{\tau}\psi) + \frac{\partial D_S}{\partial \hat{\rho}}(\partial^\nu(\bar{\psi}\psi))(\partial_\nu(\bar{\psi}\psi))\right)\right). \quad (6.15)$$

U ovim smo jednadžbama koristili notaciju  $u^\mu = (1 - v^2)^{-1/2}(1, \vec{v})$ , kojime opisujemo struju

$$\hat{\rho}u^\mu = \bar{\psi}\gamma^\mu\psi. \quad (6.16)$$

Uz izoskalarni-vektorski  $\Sigma^\mu$ , izoskalarni-skalarni  $\Sigma_S$ , izovektorski-vektorski  $\Sigma_T^\mu$  i izovektorski-skalarni  $\Sigma_{TS}$  doprinos, ovisnost funkcija o gustoći daje još dva doprinosa,  $\Sigma_{rS}$  i  $\Sigma_r^\mu$ , koji potječu od varijacije funkcije vrha po gustoći.

Gustoće odgovaraju očekivanim vrijednostima u osnovnom stanju atomske jezgre

$|\Phi\rangle$

$$\rho_S = \langle \Phi | \bar{\psi} \psi | \Phi \rangle = \rho_s^p + \rho_s^n, \quad (6.17)$$

$$\rho = \langle \Phi | \bar{\psi} \gamma^0 \psi | \Phi \rangle = \rho^p + \rho^n, \quad (6.18)$$

$$\rho_{s3} = \langle \Phi | \bar{\psi} \tau_3 \psi | \Phi \rangle = \rho_s^p - \rho_s^n, \quad (6.19)$$

$$\rho_{tv} = \langle \Phi | \bar{\psi} \tau_3 \gamma^0 \psi | \Phi \rangle = \rho^p - \rho^n. \quad (6.20)$$

## 6.2 Primjena automatske diferencijacije na funkcional DD-PC1

Funkcional “Density-dependent point coupling 1” (DD-PC1) definiran je pomoću sljedećih funkcija vrha [11]

$$G_s(\rho) = a_s + (b_s + c_s)e^{-d_s x}, \quad (6.21)$$

$$G_v(\rho) = a_v + b_v e^{-d_v x}, \quad (6.22)$$

$$G_{tv}(\rho) = b_{tv} e^{-d_{tv} x}, \quad (6.23)$$

gdje indeksi  $s$ ,  $v$  i  $tv$  označavaju izoskalar-skalar, izoskalar-vektor i izovektor-vektor kanale. Uvodimo novu varijablu,  $x$ , kao omjer,

$$x = \frac{\rho}{\rho_{sat}}, \quad (6.24)$$

gdje je  $\rho_{sat}$  nukleonska gustoća kod saturacije u simetričnoj nuklearnoj materiji. Funkcional DD-PC1 je odabran jer omogućuje izvrstan opis atomskih jezgara u cijelom periodnom sustavu [12, 13].

Računalna implementacija MBAM metode s algoritamskom diferencijacijom je testirana na pseudopodacima za koje je već bilo poznato rješenje MBAM redukcije. Testiran je model sa sedam parametara,  $a_s$ ,  $b_s$ ,  $c_s$ ,  $d_s$ ,  $a_v$ ,  $b_v$  i  $d_v$  na podacima prezentiranima u članku [14].

Pronađeno ja da nuklearnu materiju dobro opisuju gustoća saturacija  $\rho_{sat} = 0.152 \text{ fm}^{-3}$  te efektivna masa  $m_D^* = m + \Sigma_S = 0.52m$  [15].

Procedura računanja je podijeljena u dva dijela. Prvo se iterativno rješava samo-suglasna jednađžba za masu  $m^*$ , do zadovoljavajuće preciznosti

$$m_D^* = m + \Sigma_S = m + \frac{2\alpha_s(x)\rho_S}{\pi^2} m^* I_1(P_f(\rho), m^*), \quad (6.25)$$

gdje su  $I_1$  integral

$$I_1(P, a) = \int_0^P \frac{x^2 dx}{\sqrt{x^2 + a^2}}$$

i  $P_F$  Fermijev impuls

$$P_F(\rho) = \left( \frac{3\rho\pi^2}{2} \right)^{1/3}.$$

Precizan rezultat dobiva se ako se u svakome koraku masa  $m^*$  promijeni za 0.6x razlika vrijednosti dobivene jednađžbom 6.25 i vrijednosti iz prethodnog koraka.

Nakon toga se računa energija koristeći izraz

$$E_a \rho = \frac{2}{\pi^2} I_2(P_f(\rho), m^*) + m(\rho_S - \rho) + \frac{1}{2} \alpha_S(x) \rho_S^2 + \frac{1}{2} \alpha_V(x) \rho^2,$$

gdje je  $I_2$  integral

$$I_2(P, a) = \int_0^P \frac{x^4 dx}{\sqrt{x^2 + a^2}}.$$

Reziduali su

$$r_m = \frac{m_D^* - m^*(\rho_m)}{\sigma_m}$$

i

$$r_i = \frac{E_{a,i} - E_a(\rho_i)}{\sigma_e},$$

gdje su greške fiksirane na 10% vrijednosti energija i 2% vrijednosti mase. Podaci su izlistani u tablici 6.1. Samosuglasno rješenje masa i energija je pogodno za autograd jer se postupak, iako iterativan, može razložiti na kompozicije funkcija.

Rezultati integracije geodezijske jednađžbe prikazani su na slikama 6.1, 6.2 i 6.3. Problem nastaje prilikom korištenja zadane preciznosti koju preporuča scipy paket ( $rtol = 1.49012 \times 10^{-8}$ ), te je preciznost smanjena na  $rtol = 10^{-3}$ . Na donjim okvirima slika 6.1 i 6.2 vidi se da i determinanta Fisherove metrike i  $\omega$  i Riccijev skalar naglo počinju rasti za  $\tau \sim 1$ , što onemogućava pronalaženje rješenja jako velike preciznosti

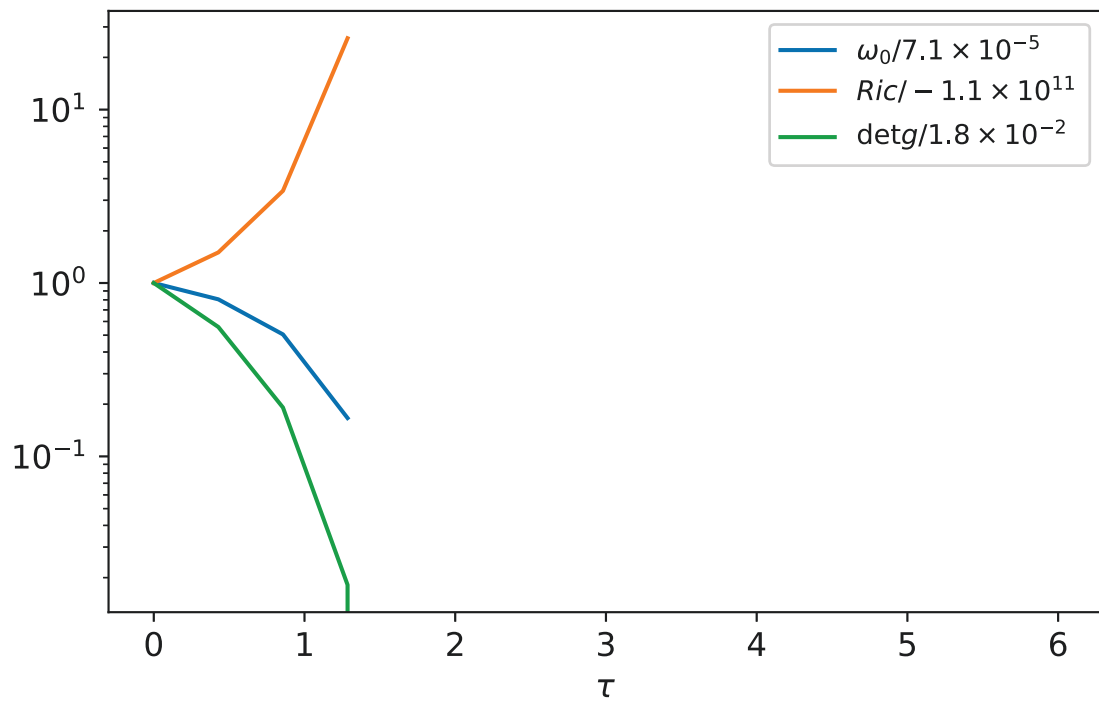
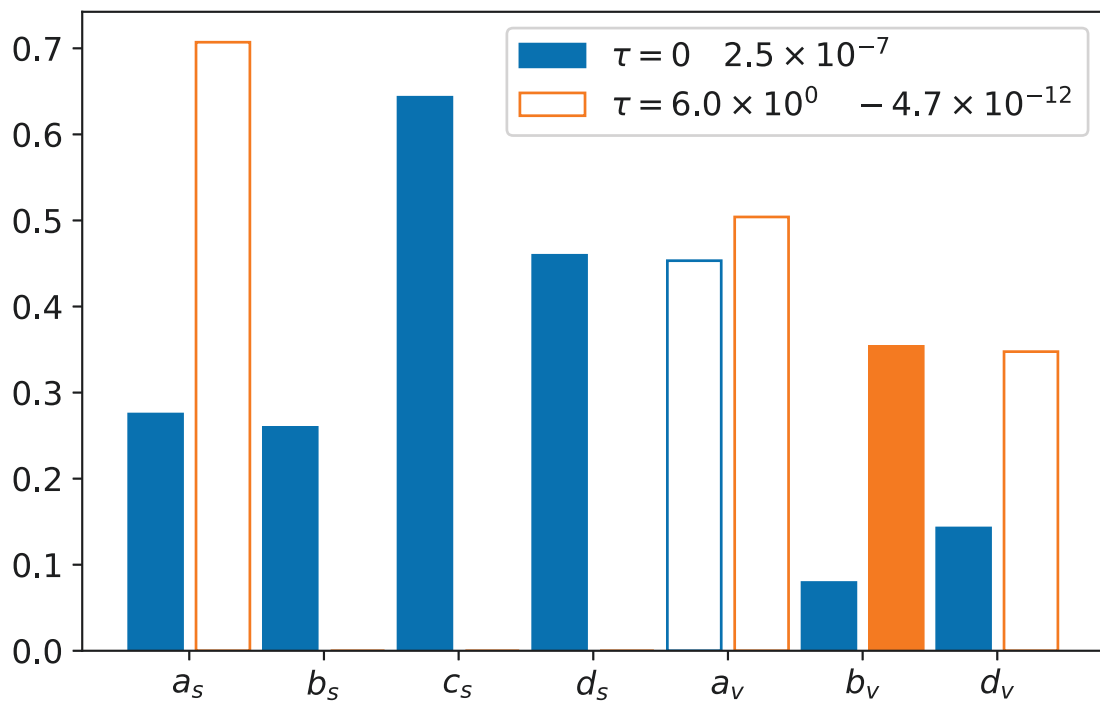
Tablica 6.1: Ovisnosti pseudopodataka energije kao funkcije gustoće korištene za testiranje MBAM metode.

$\rho$ [fm <sup>-3</sup> ]	$E_a$ [MeV]
0.04	-6.48
0.08	-12.14
0.12	-15.04
0.16	16.00
0.20	-15.09
0.24	-12.88
0.32	-5.03

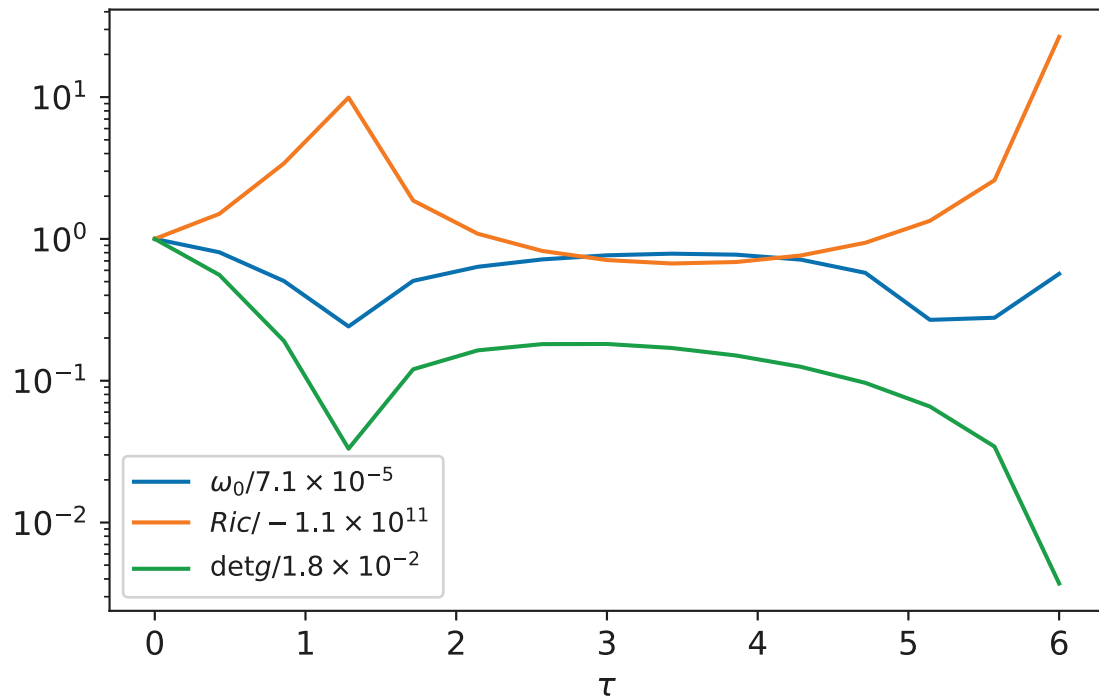
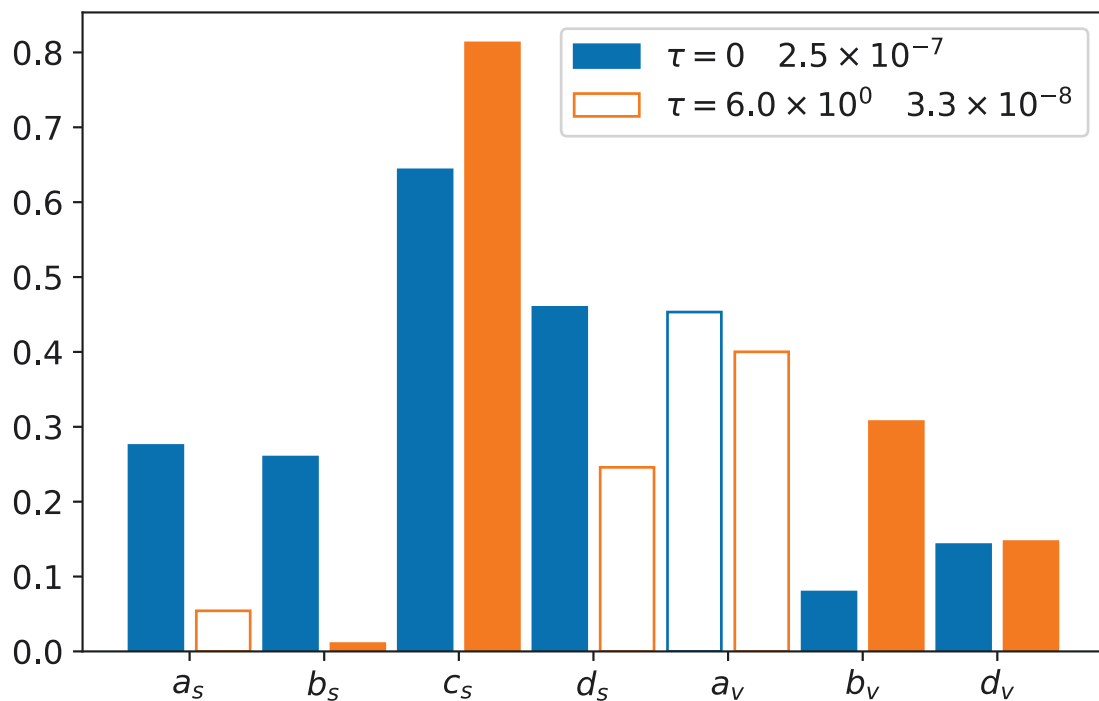
i odeint se prekida.

Nakon zadavanja manje preciznosti, integracija se nastavlja. Problem nastaje jer sustav diferencijalnih jednadžbi postaje “stiff”. Odeint je dizajniran da u tom slučaju prijeđe na drugi algoritam (BDF), no moguće je da prijelaz nije uvijek jednostavno automatski provesti. Na ovu promjenu algoritma ukazuje nagla promjena determinante metrike, što znači da su neke komponente metrike dobile veću vrijednost, a time i Christoffelovi simboli druge vrste. Ovakve promjene su indikacija prelaska u “stiff” režim [16]. Potrebno je naglasiti da nam velika preciznost niti ne treba jer parametri, prikazani na slici 6.3 ne pokazuju eksponencijalne promjene.

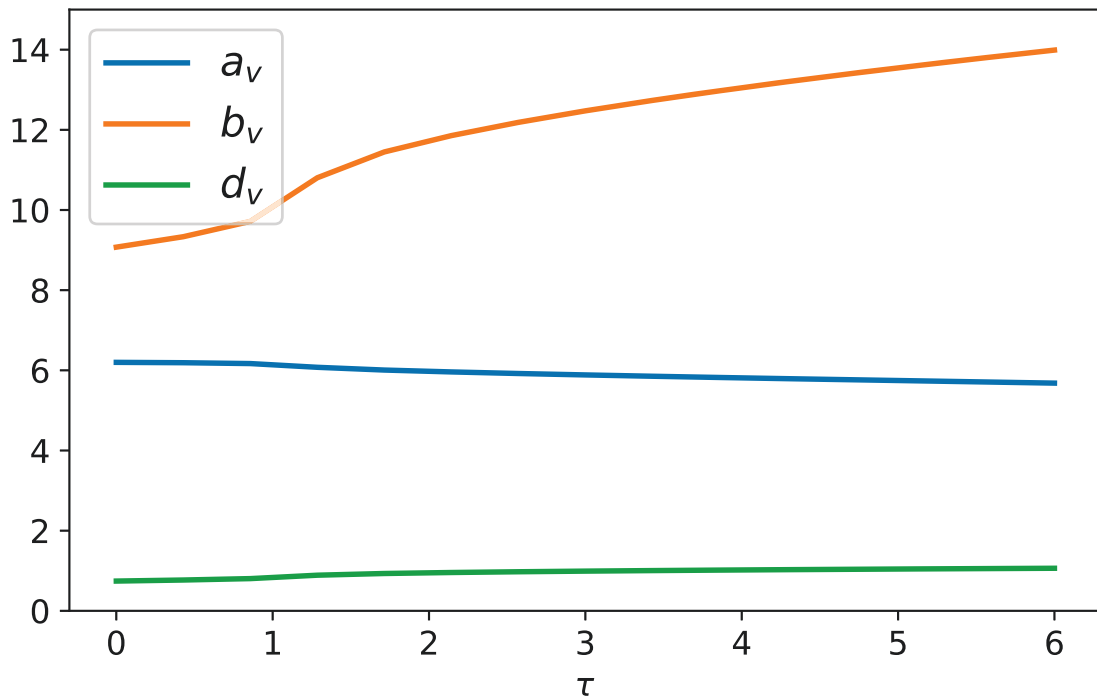
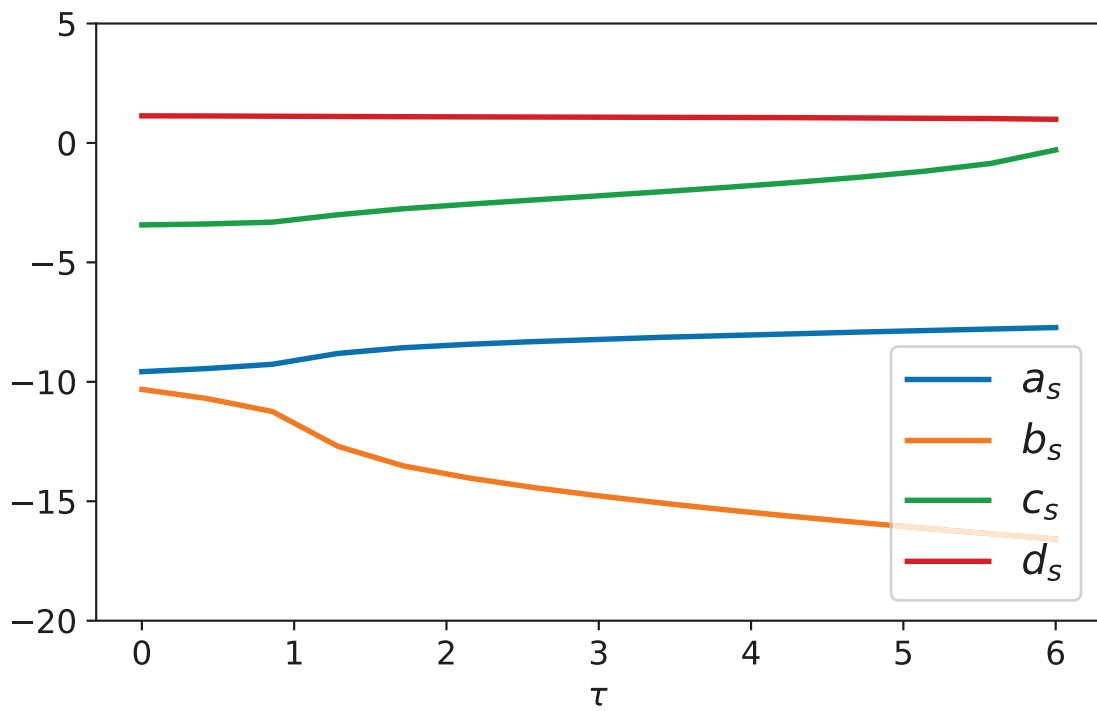




Slika 6.1: Komponente najmanjeg svojstvenog vektora dobivene za DD-PC1 funkcional.



Slika 6.2: Komponente najmanjeg svojstvenog vektora dobivene za DD-PC1 funkcional, uz odeint preciznost od  $10^{-3}$ .



Slika 6.3: Ponašanje parametara funkcionala DD-PC1 u ovisnost o parametru  $\tau$  uz preciznost od  $10^{-3}$ .

## 7 Zaključak

Modeli koji se koriste za opis većine prirodnih pojava imaju velik broj slobodnih parametara. Problemi pri određivanju vrijednosti parametara modela nastaju jer često nemamo na raspolaganju dovoljno podataka za precizno određivanje svih stupnjeva slobode modela, što dovodi do pojave da su neke kombinacije parametara modela slabo određene. Također, postoje klase modela, nazvani aljkavi modeli, kod kojih promjene određenih parametara slabo utječu na kvalitetu opisa promatrane prirodne pojave.

Ovaj diplomski rad predstavlja nastavak razvoja mogućnosti primjene koncepata informacijske geometrije u vidu razvoja aproksimacijske metode mnogostrukosti s rubom. Koristeći algoritamsku diferencijaciju, razvili smo računalnu implementaciju aproksimacijske metode mnogostrukosti s rubom koja zaobilazi ograničenja i nepreciznosti povezane s numeričkim aproksimacijama pri evaluaciji derivacija. Testirali smo mogućnosti razvijenog koda na velikom broju kompliciranih modela, od jednostavnih analitičkih prilagodbi do kaotičnih sustava te sustava s velikim intrinzičnim i ekstrinzičnim zakrivljenostima prostora parametara modela [5].

Posebno je promotren nuklearni energijski funkcional gustoće DD-PC1 za koji smo povezali numeričke probleme pri redukciji modela temeljenom na funkcionalu DD-PC1 s geometrijskim svojstvima modela - naglim promjenama u determinanti Fisherove metrike i intrinzične i ekstrinzične zakrivljenosti prostora parametara.

## Literatura

- [1] B. B. Machta, R. Chachra, M. K. Transtrum, and J. P. Sethna. Parameter Space Compression Underlies Emergent Theories and Predictive Models. *Science*, 342: 604–607, November 2013. doi: 10.1126/science.1238723.
- [2] M. K. Transtrum, B. B. Machta, K. S. Brown, B. C. Daniels, C. R. Myers, and J. P. Sethna. Perspective: Sloppiness and emergent theories in physics, biology, and beyond. *J. Chem. Phys.*, 143(1), July 2015.
- [3] M. Buchanan. Wheat from the chaff. *Nature Physics*, 11:296, April 2015. doi: 10.1038/nphys3296.
- [4] Mark K. Transtrum and Peng Qiu. Model reduction by manifold boundaries. *Phys. Rev. Lett.*, 113:098701, Aug 2014. doi: 10.1103/PhysRevLett.113.098701. URL <https://link.aps.org/doi/10.1103/PhysRevLett.113.098701>.
- [5] Benjamin Francis and Mark Transtrum. Unwinding the model manifold: choosing similarity measures to remove local minima in sloppy dynamical systems. In *APS March Meeting Abstracts*, volume 2019 of *APS Meeting Abstracts*, page X56.008, Jan 2019.
- [6] John M. Lee. *Introduction to Riemannian Manifolds*. Springer, 2018.
- [7] John M. Lee. *Introduction to Topological Manifolds*. Springer, 2011.
- [8] Mark. Fels. *An Introduction to Differential Geometry through Computation*. Springer, 2019. [https://www.researchgate.net/publication/268303006\\_An\\_Introduction\\_to\\_Differential\\_Geometry\\_through\\_Computation](https://www.researchgate.net/publication/268303006_An_Introduction_to_Differential_Geometry_through_Computation), 9.4.2019.
- [9] J. Shao. *Mathematical Statistics*. Springer, 2003.
- [10] P. Finelli, N. Kaiser, D. Vretenar, and W. Weise. Relativistic nuclear model with point-couplings constrained by QCD and chiral symmetry. *Nuclear Physics A*, 735:449–481, May 2004. doi: 10.1016/j.nuclphysa.2004.02.001.

- [11] Tamara Nikšić and Dario Vretenar. "Sloppy" nuclear energy density functionals: Effective model reduction. *Phys. Rev. C*, 94:024333, Aug 2016. doi: 10.1103/PhysRevC.94.024333. URL <https://link.aps.org/doi/10.1103/PhysRevC.94.024333>.
- [12] S. E. Agbemava, A. V. Afanasjev, D. Ray, and P. Ring. Global performance of covariant energy density functionals: Ground state observables of even-even nuclei and the estimate of theoretical uncertainties. *Phys. Rev. C*, 89:054320, May 2014. doi: 10.1103/PhysRevC.89.054320. URL <https://link.aps.org/doi/10.1103/PhysRevC.89.054320>.
- [13] S. E. Agbemava, A. V. Afanasjev, T. Nakatsukasa, and P. Ring. Covariant density functional theory: Reexamining the structure of superheavy nuclei. *Phys. Rev. C*, 92:054310, Nov 2015. doi: 10.1103/PhysRevC.92.054310. URL <https://link.aps.org/doi/10.1103/PhysRevC.92.054310>.
- [14] T. Nikšić, M. Imbrišak, and D. Vretenar. "Sloppy" nuclear energy density functionals. II. Finite nuclei. *Phys. Rev. C*, 95(5):054304, May 2017. doi: 10.1103/PhysRevC.95.054304.
- [15] T. Nikšić, D. Vretenar, and P. Ring. Relativistic nuclear energy density functionals: Mean-field and beyond. *Progress in Particle and Nuclear Physics*, 66(3): 519–548, Jul 2011. doi: 10.1016/j.pnpnp.2011.01.055.
- [16] G. Dahlquist. A special stability problem for linear multistep methods. *BIT*, 3 (27), 1963.

## A Dodatak: Python implementacija MBAM metode

U prvom potpoglavlju dan je kod kojim su dobiveni izračuni MBAM metode i pripadni grafovi, prilagođen za ispis na papiru. Originalna verzija koda, kao i primjeri iz poglavlja 5, mogu se naći na *github*-u <sup>3</sup>. Drugo potpoglavlje je dan kod korišten u poglavlju 6.

### A.1 Kod za MBAM metodu

```
# MBAM_codes.py
import autograd as au
import autograd.numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
from scipy.integrate import odeint
from scipy.integrate import ode
from scipy.integrate import solve_ivp

class solution_class:
    def __init__(self,**kwargs):
        for k,v in kwargs.items():
            if v is not None:
                self.__dict__[k] = [v]
            else:
                self.__dict__[k] = []
    def update(self,**kwargs):
        for k,v in kwargs.items():
            if v is not None:
                self.__dict__[k] = self.__dict__[k]+[v]
            else:
                self.__dict__[k] = self.__dict__[k]+[]
    def array(self):
        for k, v in self.__dict__.items():
            self.__dict__[k] = np.array(v)
```

---

<sup>3</sup><https://github.com/markoimbrisak>

```

def strip(self):
    for k, v in self.__dict__.items():
        self.__dict__[k] = v[0]
def wrap(self):
    for k, v in self.__dict__.items():
        self.__dict__[k] = [v]
class Riemann_tools:
    def __init__(self):
        pass
    def metric_inverse(self, g):
        try:
            ginv = np.linalg.inv(g)
        except:
            ginv = np.empty(g.shape)
            ginv.fill(np.nan)
        return ginv
    def metric_Gamma2(self, g, Gamma1):
        return np.einsum('ij, jab->iab',
            self.metric_inverse(g), Gamma1)
    def metric_eigenproblem(self, g):
        try:
            epr = np.linalg.eigh(g)
        except:
            epr = []
            v = np.empty(g.shape[0])
            lambda = np.empty(g.shape)
            v.fill(np.nan)
            lambda.fill(np.nan)
            epr=[v, lambda]
        return epr
    def metric_eigenvalues(self, g)->"kth_eigenvalue":
        return self.metric_eigenproblem(g)[0]
    def metric_eigenvalue(self, g, k=0)->"kth_eigenvalue":
        return self.metric_eigenproblem(g)[0][k]
    def metric_eigenvector(self, g, k=0)->"kth_eigenvalue":

```



```

    return self.metric_eigenproblem(g)[1][:,k]
def metric_determinant(self,g)->"determinant_of_g":
    return np.linalg.det(self.g)
def metric_signature(self,g)->"Metric_signature":
    return np.sum(np.sign(self.metric_eigenvalues(g)))

```

```

class embedded_manifold(Riemann_tools):

```

```

    def __init__(self):
        Riemann_tools.__init__(self)
    def embedded_metric(self,J):
        return np.einsum('im,in',J,J)
    def embedded_Gamma1(self,J,H):
        return np.einsum('im,iab->mab',J,H)
    def embedded_Gamma2(self,J,H,g):
        return self.metric_Gamma2(g,self.embedded_Gamma1(J,H))
    def normal_projection_operator(self,J,g):
        P1=np.einsum('mn,im,jn',self.metric_inverse(g),J,J)
        return 1-P1
    def embedded_Riemann_tensor(self,H,P):
        return np.einsum('ima,ij,jbn->mnab',H,P,H)\
            -np.einsum('imb,ij,jan->mnab',H,P,H)
    def embedded_Ricci_tensor(self,H,g,P):
        gi = self.metric_inverse(g)
        return np.einsum('iab,ab,ij,jmn->mn',H,gi,P,H)-\
            np.einsum('ins,ij,as,jam->mn',H,P,gi,H)
    def embedded_Ricci_scalar(self,H,g,P):
        gi = self.metric_inverse(g)
        return np.einsum('iam,ma,ij,jbn,bn',H,gi,P,H,gi)\
            -np.einsum('ims,ij,jab,ma,sb',H,P,H,gi,gi)
    def embedded_velocity2(self,g,dtheta):
        return np.einsum('m,mn,n',dtheta,g,dtheta)
    def embedded_acceleration2(self,H,P,dtheta):
        a=np.einsum('imn,ij,jab',H,P,H)
        return np.einsum('m,n,a,b,mnab',
            dtheta,dtheta,dtheta,dtheta,a)

```

```

def embedded_radius(self , H,P,g,dtheta)
    "Extrinsic_curvature_radius":
    v2=self.embedded_velocity2(g,dtheta)
    a2=self.embedded_acceleration2(H,P,dtheta)
    return v2/np.sqrt(a2)

def embedded_omega(self ,H,P,g,dtheta)->"Extrinsic_frequency":
    v2=self.embedded_velocity2(g,dtheta)
    a2=self.embedded_acceleration2(H,P,dtheta)
    return np.sqrt(np.abs(v2*a2))

def embedded_omega0(self ,H,P,g,k=0)
    "Extrinsic_frequency_in_the_k-th_eigendirection":
    v=self.metric_eigenvector(g,k)
    return self.embedded_omega(H,P,g,v)/(2*np.pi)

class diff_FIM(embedded_manifold):
    def __init__(self ,t:"x-axis_measurements",
                y:"measurements",
                sigma:"errors",
                model:"model_function"):
        self.t = t
        self.y = y
        self.f = model
        self.sigma = sigma
        self.J = au.jacobian(lambda theta:(y-model(t,theta))/sigma)
        self.H = au.hessian(lambda theta:(y-model(t,theta))/sigma)
        embedded_manifold.__init__(self)

    def _r(self , theta:"Model_parameters")->"Residuals":
        return (self.y-self.f(self.t,theta))/self.sigma

    def chi2(self , theta:"Model_parameters")->"chi^2_value":
        return np.sum((self.y-self.f(self.t,theta))**2/self.sigma**2)

    def g(self , theta:"Model_parameters"):
        "FIM_for_parameters_theta"
        return self.embedded_metric(self.J(theta))

```

```

def eigval(self , theta:"Model_parameters",k=0):
    "kth_eigenvalue_for_parameters_theta"
    return self.metric_eigenvalue(self.g(theta),k)

def eigvector(self , theta:"Model_parameters",k=0):
    "kth_eigenvalue_for_parameters_theta"
    return self.metric_eigenvector(self.g(theta),k)

def detg(self , theta:"Model_parameters")->"determinant_of_FIM":
    return self.metric_determinant(self.g(theta))

def signature(self , theta:"Model_parameters")->"Metric_signature":
    return self.metric_signature(self.g(theta))

def ginv(self , theta:"Model_parameters")->"Metric_inverse":
    return self.metric_inverse(self.g(theta))

def Gamma1(self , theta:"Model_parameters"):
    "Christoffel_symbols_of_the_first_kind"
    return self.embedded_Gamma1(self.J(theta),self.H(theta))

def Gamma2(self , theta:"Model_parameters"):
    "Christoffel_symbols_of_the_second_kind"
    return self.embedded_Gamma2(self.J(theta),
                                self.H(theta),self.g(theta))

def P(self , theta:"Model_parameters"):
    "Residual_space_normal_projection_operator"
    return self.normal_projection_operator(self.J(theta),
                                           self.g(theta))

def Riemann(self , theta:"Model_parameters")->"Riemann_tensor":
    return self.embedded_Riemann_tensor(self.H(theta),
                                         self.P(theta))

```

```

def Ricci(self , theta : "Model_parameters") -> "Ricci_tensor":
    return self.embedded_Ricci_tensor(self.H(theta),
                                       self.g(theta),
                                       self.P(theta))

def Ricci_R(self , theta : "Model_parameters"):
    "Ricci_curvature_scalar"
    return self.embedded_Ricci_scalar(self.H(theta),
                                       self.g(theta),
                                       self.P(theta))

def external_v2(self , theta : "Model_parameters",
                dtheta : "derivatives_of_model_parameters"):
    "Residual_space_velocity^2"
    return self.embedded_velocity2(self.g(theta),
                                    dtheta)

def external_a2(self ,
                theta : "Model_parameters",
                dtheta : "derivatives_of_model_parameters"):
    "Residual_space_acceleration^2"
    return self.embedded_acceleration2(self.H(theta),
                                       self.P(theta),
                                       dtheta)

def external_R(self ,
                theta : "Model_parameters",
                dtheta : "derivatives_of_model_parameters"):
    "Extrinsic_curvature_radius"
    v2 = self.external_v2(theta , dtheta)
    a2 = self.external_a2(theta , dtheta)
    return v2/np.sqrt(a2)

def external_omega(self ,
                  theta : "Model_parameters",
                  dtheta : "derivatives_of_model_parameters"):
    "Extrinsic_frequency"

```

```

return self.embedded_omega(self.H(theta),
                             self.P(theta), self.g(theta), dtheta)

def external_omegav(self,
                     theta:"Model_parameters",
                     dtheta:"derivatives_of_model_parameters"):
    "Extrinsic_normalized_frequency"
    v2=self.external_v2(theta, dtheta)
    a2=self.external_a2(theta, dtheta)
    return np.sqrt(np.abs(a2))

def calc_omega0(self,
                 theta:"Model_parameters",
                 k:"index_of_the_k-th_smallest_eigenvalue_of_g"):
    "Extrinsic_frequency_in_the_k-th_eigendirection"
    return self.embedded_omega0(self.H(theta),
                                self.P(theta),
                                self.g(theta), k)

def delete_offending_index(self,
                            tensor:"tensor_of_any_rank",
                            i:"which_dimension_to_remove"):
    "minor_of_the_tensor"
    Naxis = np.size(tensor.shape)
    NewT = tensor
    for j in range(Naxis):
        NewT = np.delete(NewT, i, axis=j)
    return NewT

def find_offending_index(self, g:"Metric_tensor"):
    """
    Finds minor of FIM without
    which the FIM has a full rank
    """

```

```

N      = g.shape[0]
cut    = np.array([
        np.linalg.matrix_rank(self.delete_offending_index(g,i))
        for i in range(N)])==(N-1)
if np.sum(cut)==0:
    raise Warning('Nothing_found_by_removing_1_axis')
return np.argmax(cut)

def find_MBAM_IC(self ,
                  theta:"Model_parameters",
                  k:"Index_of_the_eigendirection"=0):
    "Produces_initial_conditions_for_the_geodesic_equation"
    g = self.g(theta)
    N = g.shape[0]
    if np.linalg.matrix_rank(g)==N:
        lambda,v    = self.metric_eigenproblem(g)
        V = v[:,k]
        theta2    = np.einsum('i,j,ij',theta,theta,g)
        v2    = np.einsum('i,j,ij',V,V,g)
    elif k<N-1:
        i = self.find_offending_index(g)
        gnew = self.delete_offending_index(g,i)
        thetanew = self.delete_offending_index(theta,i)
        lambda,v    = np.linalg.eigh(gnew)
        theta2    = np.einsum('i,j,ij',thetanew,thetanew,gnew)
        v2    = np.einsum('i,j,ij',v[:,k],v[:,k],gnew)
        v = v[:,k]
        V = np.append(v[:i],[0])
        V = np.append(V,v[i:])
    else:
        i    = self.find_offending_index(g)
        V    = np.zeros(N)
        V[i] = 1
        gnew = self.delete_offending_index(g,i)
        thetanew = self.delete_offending_index(theta,i)

```

```

        lambda, v = self.metric_eigenproblem(gnew)
        theta2 = np.einsum('i, j, ij', thetanew, thetanew, gnew)
        v2 = np.einsum('i, j, ij', v[:,0], v[:,0], gnew)
        tau = np.sqrt(theta2/v2)
        return np.append(theta, V), tau

def MBAM_RHS(self, V: "2N-dimensional_initial_conditions_vector")
    "RHS_of_the_geodesic_equation":
    N = int(np.size(V)/2)
    theta = V[:N]
    dtheta = V[N:]
    g = self.g(theta)
    ret = np.array([dtheta])
    if np.linalg.matrix_rank(g)==N:

        ret = np.append(
            ret,
            -np.einsum('a, b, cab->c',
                dtheta, dtheta, self.Gamma2(theta)))
    else:
        return np.array(2*N*[np.nan])

    return ret

def MBAM_jac_RHS(self,
    V: "2N-dimensional_initial_conditions_vector"):
    "RHS_of_the_geodesic_equation"
    N = int(np.size(V)/2)
    theta = V[:N]
    dtheta = V[N:]
    g = self.g(theta)
    ret = np.c_[np.zeros((N,N)), np.diag(N*[1])]
    if np.linalg.matrix_rank(g)==N:
        ret1 = np.c_[np.zeros((N,N)),
            -2*np.einsum('a, bi, cab->ci',

```

```

        dtheta ,
        np.diag(N*[1]),
        self.Gamma2(theta))]
    ret = np.r_[ret,ret1]
else:
    return np.array(2*N*[np.nan])
return ret

def statistics(self,theta):
    y = self.f(self.t,theta)
    J = self.J(theta)
    g = self.g(theta)
    Gamma2 = self.Gamma2(theta)
    omega0 = self.calc_omega0(theta,0)
    R = self.Ricci_R(theta)
    return y, J, g, Gamma2, omega0, R

def run_MBAM(self,
    theta:"Model_parameters",
    k:"Initial_eigendirection"=0,
    T=None,
    use_dfun=False,**odekwargs):
    "computes_the_geodesic_equation"
    def fun(V,t):
        return self.MBAM.RHS(V)
    def funD(V,t):
        return self.MBAM.jac.RHS(V)
    V0,tau = self.find_MBAM.IC(theta,k)
    N_parameters=int(np.size(V0)/2)
    if T is None:
        T = np.linspace(0,10*tau,100)
        constructed_tau = True
    else:
        constructed_tau = False
    sols = solution_class(y=None,J=None, g=None,

```



```

        Gamma2=None,
        omega0=None,
        R=None,
        tau=None,
        theta=None,
        dtheta=None, detg=None)
    if use_dfun:
        S = odeint(fun, V0, T, Dfun=funD, **odekwargs)
    else:
        S = odeint(fun, V0, T, **odekwargs)
    i = 0
    while i < len(T):
        V = S[i]

        y, J, g, Gamma2, omega0, R=\
            self.statistics(V[:N_parameters])
        sols.update(y=y, J=J,
                    g=g, Gamma2=Gamma2,
                    omega0=omega0, R=R,
                    tau=T[i],
                    theta=V[:N_parameters],
                    dtheta=V[N_parameters:],
                    detg=np.linalg.det(g))

        i+=1
    sols.array()
    if constructed_tau:
        return tau, sols
    else:
        return sols

```

```

def SymLogNorm(dg:"Values to plot w/ pcolormesh"):
    "Matplotlib lognorm"
    adg = np.abs(dg)
    vmax = adg.max()

```

```

vmin = adg.min()
vmin = np.max([10**(-5.)*vmax, vmin])
if np.sum(dg<0)>0:
    return colors.SymLogNorm(linthresh=vmin,
                              vmin=-vmax, vmax=vmax)
else:
    return colors.LogNorm(vmin=vmin, vmax=vmax)

```

```

class MBAM_plotting(diff_FIM):
    def __init__(self,
                 model:"Model_to_analyze",
                 NAME:"string",
                 X:"X_data", Y:"Y_data", sigmaY:"Error_bars",
                 theta_bf:"Best_fitting_parameters",
                 xlim:"x_values_limit", ylim:"y_values_limit"):
        self.NAME = NAME
        self.xlim = xlim
        self.ylim = ylim
        self.ND = len(X)
        self.NP = len(theta_bf)
        diff_FIM.__init__(self, X, Y, sigmaY, model)
        self.theta_bf = theta_bf

    def plot_data(self, ax:"matplotlib_axis",
                 color:"color_of_data_points"='C0',
                 fit:"show_a_fit"=True):
        """points_and_residuals_plotted"""
        ax[0].set_xlim(self.t.min(), self.t.max())
        ax[1].set_xlim(self.t.min(), self.t.max())
        ax[0].set_ylim(self.y.min(), self.y.max())
        ax[1].set_ylim(-2,2)
        ax[0].errorbar(self.t,
                      self.y,
                      yerr=self.sigma,
                      fmt=color+'o', capsize=2)

```

```

if fit:
    ax[0].plot(self.t,
               self.f(self.t, self.theta_bf),
               'k', lw=2)
ax[0].set_xlabel('$t$')
ax[0].set_ylabel('$y(t)$')
ax[1].errorbar(self.t,
               self.r(self.theta_bf),
               yerr=self.ND*(1,),
               fmt='o', capsize=2)
ax[1].axhline(0, color='k')
ax[1].set_xlabel('$t$')
ax[1].set_ylabel('$r(t)$')

```

```

def plot_scalar(self,
                 ax:"Matplotlib_axis",
                 X:"X_mesh_to_plot",
                 Y:"Y_mesh_to_plot",
                 Z:"Colors_to_plot",
                 Vx:"X_components_of_vectors_to_plot"=None,
                 Vy:"Y_components_of_vectors_to_plot"=None,
                 xlabel=r'$\theta^0$',
                 ylabel=r'$\theta^1$',
                 veccolor='k'):
    ax.plot(*self.theta_bf, 'ro')
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    im=ax.pcolormesh(X,Y,Z,norm=SymLogNorm(Z.flatten()),
                    cmap='gray')
    plt.colorbar(im, ax=ax)
    ax.set_xlim(*self.xlim)
    ax.set_ylim(*self.ylim)
    if Vx is not None:
        ax.streamplot(X,Y,Vx,Vy, color=veccolor)

```

```

def construct_mesh(self ,N:"Produces a mesh of dimension N"):
    """List of x and y points and the corresponding meshgrid"""
    theta1 = np.linspace(*self.xlim ,N)
    theta2 = np.linspace(*self.ylim ,N)
    X,Y = np.meshgrid(theta1 ,theta2)
    return theta1 , theta2 , X, Y
def apply_on_mesh(self ,
                  N:"Mesh dimension" ,
                  F:"List of functions to plot"):
    """parameters , mesh and evaluations of functions on the mesh"""
    theta1 , theta2 , X, Y = self.construct_mesh(N)
    return theta1 , theta2 , X, Y,\
           np.array([[ f(np.array([t1 ,t2]))
                       for f in F] for t1 in theta1]
                    for t2 in theta2])

def bar_plot(self ,
             ax:"Axis to plot on" ,
             *args:"Vectors to plot" ,
             xlabel='', ylabel='',
             plabels:"x tick labels"=None,
             labels:"Labels of vectors"=None,
             colors:"Vector colors"=None, lw=1):
    """
    Adds a plot of eigenvectors as bar plots ,
    filled if a component is positive , white if negative
    """
    Nv = len(args)
    bw = .5/Nv
    for i in range(Nv):
        v = args[i]
        if labels is not None and i<len(labels):
            label = labels[i]
        else:
            label = ""

```

```

    if colors is not None and i<len(colors):
        color = colors[i]
    else:
        color = "k"
    Ntheta = np.size(v)
    ax.bar(np.arange(Ntheta)+bw,np.abs(v),.8/Nv,
           color=np.where(v>0,color,"1.0"),
           edgecolor=color,label=label,lw=1)
    bw+=1/Nv
ax.set_xticks(np.arange(0.5,Ntheta+.5,1))
#ax.set_yscale('log')
if plabels is not None:
    ax.set_xticklabels(plabels)
else:
    ax.set_xticklabels([r"$\theta^{ "+ "%d"%i+ }$"
                        for i in range(Ntheta)])
if labels is not None:
    ax.legend()

```

```

class MBAM.odeint(embedded_manifold):
    def __init__(self, function,
                 T, IC, N_parameters,
                 N_equations, **odeint_kwargs):
        self.T = T
        self.IC = IC
        self.function = function
        self.N_parameters = N_parameters
        self.N_equations = N_equations
        self.odeint_kwargs = odeint_kwargs
        embedded_manifold.__init__(self)

    def find_solutions(self, theta, full_output=True):

```

```

if 'full_output' in self.odeint_kwargs.keys():
    K, fo=odeint(self.function ,
                self.IC,
                self.T,
                args=(theta ,) ,
                **self.odeint_kwargs)
else :
    K=odeint(self.function ,
            self.IC,
            self.T,
            args=(theta ,) ,
            **self.odeint_kwargs)
y, J, H = self._construct_yJH(K)
g = self.embedded_metric(J)
Gamma2 = self.embedded_Gamma2(J,H,g)
if full_output:
    P = self.normal_projection_operator(J, g)
    omega0 = self.embedded_omega0(H,P,g)
    R = self.embedded_Ricci_scalar(H, g, P)
    return y, J, g, Gamma2, omega0, R
else :
    return Gamma2

```

```

def _construct_H_from_triu(self ,h):

```

```

    H = np.zeros((self.N_parameters , self.N_parameters))
    H[np.triu_indices(self.N_parameters)]=h
    H = H+H.T-np.diag(H)
    return H

```

```

def _construct_J(self ,K):

```

```

    J = K[:, self.N_equations : self.N_equations*(1+ self.N_parameters)]
    j = np.array([[j[k:: self.N_equations]
                  for k in range(self.N_equations)] for j in J])

```

```

        return -np.vstack([j[:,i,:] for i in range(self.N_equations)])

def _construct_H(self,K):
    H = K[:,self.N_equations*(1+self.N_parameters):]
    h = np.array([[self._construct_H_from_triu(h[k::self.N_equations])
                    for k in range(self.N_equations)] for h in H])
    return -np.vstack([h[:,i,:,:] for i in range(self.N_equations)])

def _construct_yJH(self,K):
    y = K[:,self.N_equations]
    j = self._construct_J(K)
    h = self._construct_H(K)
    return y, j, h

def rhs(self,t,V):
    theta = V[:self.N_parameters]
    dtheta = V[self.N_parameters:]
    Gamma2 = self.find_solutions(theta,False)
    return np.append(theta,-np.einsum('abc,b,c',Gamma2,dtheta,dtheta))

def rhs_odeint(self,V,t):
    theta = V[:self.N_parameters]
    dtheta = V[self.N_parameters:]
    Gamma2 = self.find_solutions(theta,False)
    return np.append(theta,-np.einsum('abc,b,c',Gamma2,dtheta,dtheta))

def run_MBAM(self,IC,T,do_odeint=True,**odeint_kwargs):
    """computes the geodesic equation"""
    self.times = T

    sols = solution_class(y=None,J=None,g=None,
                           Gamma2=None,omega0=None,R=None,
                           tau=None,theta=None,
                           dtheta=None,detg=None)

    if do_odeint:
        S = odeint(self.rhs_odeint,IC,T,**odeint_kwargs)

```

```

#S = solve_ivp(self.rhs_odeint ,
               [0,np.max(T)],
               IC,t_eval=T,
               **odeint_kwargs)

i = 0
while i < len(T):
    V = S[i]
    y, J, g, Gamma2, omega0, R=\
        self.find_solutions(V[:self.N_parameters])
    sols.update(y=y, J=J, g=g, Gamma2=Gamma2,
               omega0=omega0, R=R, tau=T[i],
               theta=V[:self.N_parameters],
               dtheta=V[self.N_parameters:],
               detg=np.linalg.det(g))
    i+=1
else:
    r = ode(self.rhs)
    r.set_integrator('vode', method='bdf')
    r.set_initial_value(IC, 0)
    i = 0
    SL = []
    while r.successful() and i < len(T):
        V =r.integrate(T[i])
        y, J, g, Gamma2, omega0, R=\
            self.find_solutions(V[:self.N_parameters])
        sols.update(y=y, J=J, g=g, Gamma2=Gamma2,
                   omega0=omega0, R=R, tau=T[i],
                   theta=V[:self.N_parameters],
                   dtheta=V[self.N_parameters:],
                   detg=np.linalg.det(g))
        print(T[i], V[:self.N_parameters])
        i+=1
    sols.array()
    return sols
def plot_data(self ,ax,T,ys ,rs ,y):

```



```

for i in range(ys.shape[1]):
    ax[0].errorbar(T, ys[:, i], yerr=1, fmt='o', capsize=2)
    ax[1].errorbar(T, rs[:, i], yerr=1, fmt='o', capsize=2)

ax[0].plot(T, y, 'k')
ax[0].set_xlabel('$t$')
ax[0].set_ylabel('$y(t)$')
ax[1].axhline(0, color='k')
ax[1].set_xlabel('$t$')
ax[1].set_ylabel('$r(t)$')
def bar_plot(self, *args, **kwargs):
    MBAM_plotting.bar_plot(None, *args, **kwargs)
def splitplots(N, figsize):
    Fs = [plt.subplots(1,1,figsize=figsize) for i in np.arange(N)]
    ax = [F[1] for F in Fs]
    F = [F[0] for F in Fs]
    return F, ax

```

## A.2 Kod za MBAM metodu na funkcionalu DDPC-1

```
# DDPC-1.ipynb
%run "MBAM_codes.py"
import autograd.numpy as np
import autograd as au
hqc = 197.328284
amu=939.0/hqc
xmix=0.6
maxi=100

npar1=4
npar2=3
npar=npar1+npar2

dsat = 0.152

param=np.array([-10.0462,-9.1504,-6.4273,1.3724,5.9195,8.8637,0.6584])

rhoam = np.array([0.152])
amst =np.array([0.58])
damst = amu*2.0*abs(amst)/100
#DD-PC1 values
rhoea = np.array([0.04,0.08,0.12,0.16,0.20,0.24,0.32])
#DD-PC1 values
ea = np.array([-6.48,-12.13,-15.04,-16.00,-15.09,-12.88,-5.03])
dea = 10.0*abs(ea)/100

rho_all = np.array([0.04,0.08,0.12,0.16,0.20,0.24,0.32,0.40,0.48,\
                    0.56,0.64,0.80,0.96])
ea_all = np.array([-6.48,-12.13,-15.04,-16.00,-15.09,-12.88,-5.03,\
                    2.13,15.46,34.39,58.35,121.25,204.02])

nmax1 = len(rhoam)
nmax2 = len(rhoea)
```

```
nmax = nmax1+nmax2
```

```
def rint1(xlo,xup,a):
```

```
#evaluates the integral \int_{xlo}^{xup} \{\frac{x^2 dx}{\sqrt{x^2+a^2}}\}
```

```
fup1 = 0.5*xup*np.sqrt(a**2+xup**2)
```

```
fup2 = -0.5*a**2*np.log(np.sqrt(a**2+xup**2)+xup)
```

```
flo1 = 0.5*xlo*np.sqrt(a**2+xlo**2)
```

```
flo2 = -0.5*a**2*np.log(np.sqrt(a**2+xlo**2)+xlo)
```

```
result = fup1+fup2-flo1-flo2
```

```
return result
```

```
def rint2(xlo,xup,a):
```

```
#evaluates the integral \int_{xlo}^{xup} \{\frac{x^4 dx}{\sqrt{x^2+a^2}}\}
```

```
fup1 = 0.125*xup*np.sqrt(a**2+xup**2)*(2*xup**2-3*a**2)
```

```
fup2 = 3*0.125*a**4*np.log(xup+np.sqrt(xup**2+a**2))
```

```
flo1 = 0.125*xlo*np.sqrt(a**2+xlo**2)*(2*xlo**2-3*a**2)
```

```
flo2 = 3*0.125*a**4*np.log(xlo+np.sqrt(xlo**2+a**2))
```

```
result = fup1+fup2-flo1-flo2
```

```
return result
```

```
def rint3(xlo,xup,a):
```

```
#evaluates the integral \int_{xlo}^{xup} \{\frac{x^2 dx}{(x^2+a^2)^{3/2}}\}
```

```
fup1 = np.log(np.sqrt(a**2+xup**2) + xup)
```

```
fup2 = -xup/np.sqrt(xup**2+a**2)
```

```
flo1 = np.log(np.sqrt(a**2+xlo**2) + xlo)
```

```
flo2 = -xlo/np.sqrt(xlo**2+a**2)
```

```
result = fup1+fup2-flo1-flo2
```

```
return result
```

```
def rint4(xlo,xup,a):
```

```
#evaluates the integral \int_{xlo}^{xup} \{\frac{x^2 dx}{(x^2+a^2)^{5/2}}\}
```

```
fup = xup**3/(3*a**2*(a**2+xup**2)**1.5)
```

```
flo = xlo**3/(3*a**2*(a**2+xlo**2)**1.5)
```

```
result = fup-flo
```

```
return result
```

```
def rint5(xlo, xup, a):
```

```
#evaluates the integral \int_{xlo}^{xup} \frac{x^4 dx}{(x^2+a^2)^{3/2}}
```

```
fup1 = (3*a**2*xup+xup**3)/np.sqrt(a**2+xup**2)
```

```
fup2 = -3*a**2*np.log(np.sqrt(a**2+xup**2) + xup)
```

```
flo1 = (3*a**2*xlo+xlo**3)/np.sqrt(a**2+xlo**2)
```

```
flo2 = -3*a**2*np.log(np.sqrt(a**2+xlo**2) + xlo)
```

```
result = 0.5*(fup1+fup2-flo1-flo2)
```

```
return result
```

```
def rint6(xlo, xup, a):
```

```
#evaluates the integral \int_{xlo}^{xup} \frac{x^4 dx}{(x^2+a^2)^{5/2}}
```

```
fup1 = -(3*a**2+4*xup**2)*xup/3/np.sqrt(a**2+xup**2)**3
```

```
fup2 = np.log(np.sqrt(a**2+xup**2) + xup)
```

```
flo1 = -(3*a**2+4*xlo**2)*xlo/3/np.sqrt(a**2+xlo**2)**3
```

```
flo2 = np.log(np.sqrt(a**2+xlo**2) + xlo)
```

```
result = fup1+fup2-flo1-flo2
```

```
return result
```

```
def nucmat(param, rv):
```

```
a_s = param[0]
```

```
b_s = param[1]
```

```
c_s = param[2]
```

```
d_s = param[3]
```

```

a_v = param[4]
b_v = param[5]
c_v = 0.0
d_v = param[6]

x = rv/dsat
alps = a_s+(b_s+c_s*x)*np.exp(-d_s*x)
alpv = a_v+(b_v+c_v*x)*np.exp(-d_v*x)

pf = (3*rv*np.pi**2/2)**(1./3.)

amstar = amu
conv=0.0
for i in range(1,maxi,1):
    amstar=amstar+xmix*conv
    rs = (2.0/np.pi**2)*amstar*rint1(0.0,pf,amstar)

    amstar_new = amu+alps*rs
    conv = amstar_new-amstar
    if abs(conv)<=1.e-13:
        break
if abs(conv) > 1.e-13:
    print( rv , conv , 'no convergence ')

ea = 2*rint2(0.0,pf,amstar)/np.pi**2 \
+amu*(rs-rv)+0.5*alps*rs**2+0.5*alpv*rv**2

ea = hqc*ea/rv

return [amstar,ea]

```

```

def chi(param):
    chi=0.0
    for m in range(0,nmax1,1):
        rv = rhoam[m]

```

```

    amth=nucmat(param,rv)[0]/amu
    sig = damst[m]
    amemp=amst[m]
    chi=chi+((amth-amemp)/sig)**2

    for m in range(0,nmax2,1):
        rv = rhoea[m]
        eath=nucmat(param,rv)[1]
        sig = dea[m]
        eaemp=ea[m]
        chi=chi+((eath-eaemp)/sig)**2
    return chi
def function(param):
    r=[]
    for m in range(0,nmax1,1):
        rv = rhoam[m]
        amth=nucmat(param,rv)[0]/amu
        sig = damst[m]
        r.append(amth)
    for m in range(0,nmax2,1):
        rv = rhoea[m]
        eath=nucmat(param,rv)[1]
        sig = dea[m]
        eaemp=ea[m]
        r.append(eath)
    return np.array(r)

from scipy import optimize
par_bf=param

print ('Optimizing with Nelder-Mead... ')
result = optimize.minimize(chi,par_bf,method='Nelder-Mead',\
options={'maxiter':5000,'maxfev':9000})
print(result)
par_bf = result.x

```

```

print( 'par_bf=', par_bf)

sigmas=np.append(damst, dea)
y      = np.append(amst, ea)
class model(MBAM_plotting):
    def __init__(self, *args, **kwargs):
        MBAM_plotting.__init__(self, self._model, 'DDPC1', *args, **kwargs)
        self._title = r'DD-PC1'
    def _model(self, T, theta):
        return function(theta)

T=np.arange(nmax1+nmax2)
MPL = model(T, y, sigmas, par_bf, xlim=[-2,2], ylim=[0,3])
MPL.MBAM = MPL.run_MBAM(par_bf, k=0,
                        T=np.linspace(0, 6., 15),
                        use_dfun=True, rtol=0.001)

def format(x):
    exp=np.floor(np.log10(np.abs(x)))
    return r"%1f\times 10^{%d}"%(x/10**exp, exp)

f, ax = splitplots(4, (4, 4))
axs=ax[:2]
bf=np.linalg.eigh(MPL.g(par_bf))
MPL.bar_plot(axs[0],
             np.linalg.eigh(MPL.g(MPL.MBAM.theta[0]))[1][0,:],
             np.linalg.eigh(MPL.g(MPL.MBAM.theta[-1]))[1][0,:],
             plabels=[r'$a_s$', r'$b_s$', r'$c_s$', r'$d_s$',
                    r'$a_v$', r'$b_v$', r'$d_v$'],
             labels=[r"$\tau=0\quad"+format(np.linalg.eigh(MPL.g(
                                                         MPL.MBAM.theta[0]))[0][0])+ '$',
                    r"$\tau="+format(MPL.MBAM.tau[-1])+ '\quad'+\
                    format(np.linalg.eigh(MPL.g(MPL.MBAM.theta[-1]))[0][0])+ '$'],
             colors=["C0", "C1"])
axs[1].plot(MPL.MBAM.tau, MPL.MBAM.omega0/MPL.MBAM.omega0[0],

```

```

        label=r"$\omega_0/"+format(MPL.MBAM.omega0[0])+"$"
    axs[1].plot(MPL.MBAM.tau,MPL.MBAM.R/MPL.MBAM.R[0],
        label=r"$Ric/"+format(MPL.MBAM.R[0])+"$")
    axs[1].plot(MPL.MBAM.tau,MPL.MBAM.detg/MPL.MBAM.detg[0],
        label=r"$\det_g/"+format(MPL.MBAM.detg[0])+"$")
    axs[1].set_xlabel(r"$\tau$")
    axs[1].legend()
    axs[1].set_yscale("log")
    axs=ax[2:]
    for i,l in enumerate([r'$a_s$',r'$b_s$',r'$c_s$',r'$d_s$']):
        axs[0].plot(MPL.MBAM.tau,MPL.MBAM.theta.T[i],label=l,lw=2)
    for i,l in enumerate([r'$a_v$',r'$b_v$',r'$d_v$']):
        axs[1].plot(MPL.MBAM.tau,MPL.MBAM.theta.T[4+i],label=l,lw=2)
    for ax in axs:
        ax.legend(fontsize=15)
        ax.set_xlabel(r"$\tau$")
    axs[1].set_ylim(0,15)
    axs[0].set_ylim(-20,5)
    for i,f in enumerate(Fs):
        f.tight_layout()
        f.savefig(MPL._title+'_%d.pdf'%i)

```