

Poluglatka Newtonova metoda u Banachovim prostorima

Gunja, Marin

Master's thesis / Diplomski rad

2020

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:217:889405>

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

Download date / Datum preuzimanja: **2024-07-27**



Repository / Repozitorij:

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



Poluglatka Newtonova metoda u Banachovim prostorima

Gunja, Marin

Master's thesis / Diplomski rad

2020

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:217:889405>

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

Download date / Datum preuzimanja: **2024-06-18**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Marin Gunja

POLUGLATKA NEWTONOVA METODA
U BANACHOVIM PROSTORIMA

Diplomski rad

Voditelj rada:
izv. prof. dr. sc. Marko Vr-
doljak

Zagreb, rujan 2020.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

*Obitelji, prijateljima, kolegama, i svima ostalima koji su me nebrojeno puta pitali hoću li
stići dovršiti ovaj rad.*

Sadržaj

Sadržaj	iv
Uvod	1
1 Glatki realni slučaj	2
1.1 Motivacija i metoda	2
1.2 Dokaz konvergencije, ocjena greške, kriterij zaustavljanja za jednostruke nultočke	6
1.3 Newtonova metoda i višestruke nultočke	10
1.4 Implementacija u Pythonu i neki primjeri	11
2 Glatki realni slučaj, poopćenje	16
2.1 Opis metode i uvodni primjer	16
2.2 Lokalna konvergencija Newtonove metode za glatke sustave	20
2.3 Globalna konvergencija	26
2.4 Newtonova metoda i bezuvjetna optimizacija	27
2.5 Implementacija u Pythonu	28
3 Generalizacija Newtonove metode	30
3.1 Uvod, generalizirani diferencijal i generalizirana Newtonova metoda	30
3.2 Poluglatkoća, teoremi konvergencije poluglatke Newtonove metode	35
3.3 Poluglatka Newtonova metoda u Banachovim prostorima	40
3.4 Primjeri — numeričko rješavanje ODJ	41
3.5 Konveksni slučaj	58
Bibliografija	61

Uvod

Često se u radu susrećemo s problemima traženja nultočki funkcija. Točnije, često imamo funkciju $F: X \rightarrow Y$ (zahtjeve na skupove X i Y ćemo specificirati kroz ovaj rad) i tražimo $x_* \in X$ takav da $F(x_*) = 0$. Na primjer, problem aproksimacije $\sqrt{2}$ možemo svesti na problem traženja aproksimacije pozitivne nultočke funkcije $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x^2 - 2$.

Za početak se bavimo funkcijama $F: A \rightarrow \mathbb{R}^n$, gdje je $A \subseteq \mathbb{R}^n$, koje su „dovoljno glatke”. Najjednostavniji slučaj je kada je A neki interval u \mathbb{R} , a $n = 1$. Tada imamo realnu funkciju realne varijable. Proučit ćemo Newtonovu metodu u ovom slučaju, iskazati i dokazati teoreme konvergencije, dati ocjenu greške i brzine konvergencije. Nakon toga ćemo se pozabaviti slučajem kada je A neki otvoreni podskup od \mathbb{R}^n , a n ne mora biti 1. Naime, često se susrećemo sa sustavima nelinearnih jednadžbi čija rješenja trebamo pronaći, a taj problem prirodno svedemo na problem traženja nultočki vektorskih funkcija.

Nakon toga se isplati proučavati $F: A \rightarrow \mathbb{R}^n$, gdje je $A \subseteq \mathbb{R}^n$ otvoren, a F ne mora nužno biti glatka. Uvest ćemo svojevrsta poopćenja diferencijala i poopćenje Newtonove metode.

Glavni cilj ovoga rada je poopćiti Newtonovu metodu tako da ju možemo koristiti za nalaženje nultočki takozvanih *poluglatkih* funkcija čija domena X i kodomena Y su Banachovi prostori.

Bit će dani i Python kodovi. Odlučili smo se za Python jer ima odličan paket za simboličko računanje (SymPy) koji se odlično slaže s paketima za crtanje grafova i numeričkim paketima.

Poglavlje 1

Glatki realni slučaj

1.1 Motivacija i metoda

Neka je dana $f: I \rightarrow \mathbb{R}$, gdje je $I \subseteq \mathbb{R}$ neki interval. Želimo naći $x_* \in I$ takav da $f(x_*) = 0$, tj. nultočku funkcije f . Za sada pretpostavljamo samo da je f neprekidna i da ima izolirane nultočke. Recimo da imamo neku lokalnu aproksimaciju za nultočku i označimo tu aproksimaciju s x_0 . Ideja je f „zamijeniti” nekom jednostavnijom funkcijom kojoj znamo naći nultočke, ali koja i dalje „dobro” aproksimira f oko x_0 .

Pretpostavimo sada da je f derivabilna u x_0 . Tada znamo da postoji tangenta na graf funkcije f u točki $(x_0, f(x_0))$. Također, f možemo aproksimirati funkcijom $g(x) = f'(x_0)(x - x_0) + f(x_0)$ (uočimo: graf funkcije g je upravo prethodno spomenuta tangenta). Ovakvu funkciju g još nazivamo i *linearnom aproksimacijom od f u točki x_0* . No, g je linearna funkcija, te njenu nultočku, ako postoji, lako nađemo. Primijetimo da je nužan i dovoljan uvjet za postojanje nultočke od g da je $f'(x_0) \neq 0$ ili da je $f(x_0) = 0$. Ali, drugi slučaj nam i nije pretjerano važan, jer to upravo znači da je x_0 nultočka funkcije f i nemamo više što aproksimirati. Označimo nultočku funkcije g s x_1 i za novu aproksimaciju nultočke funkcije f uzmemo upravo x_1 . Naravno, pitanje je je li to dovoljno dobra aproksimacija i je li ta aproksimacija uopće bolja od prethodne (preciznije, x_0), ali do toga ćemo još doći.

Konačno, pretpostavimo sada da je f derivabilna na cijelom I (svojoj domeni), da svaka lokalna aproksimacija od f ima nultočku, i da su te nultočke sadržane u I . Neka je x_0 početna aproksimacija nultočke od f . Definiramo $g_0(x) = f'(x_0)(x - x_0) + f(x_0)$. Njena nultočka, x_1 , je dana s

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \quad (1.1)$$

Sada nam je x_1 nova aproksimacija nultočke od f . Definiramo $g_1(x) = f'(x_1)(x - x_1) + f(x_1)$

i izračunamo njenu nultočku, koja je dana s

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}. \quad (1.2)$$

Induktivno, definiramo niz aproksimacija nultočke od f s

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (1.3)$$

Napomena 1.1.1. U praksi obično ne provjeravamo uvjet vezan za nultočke linearnih aproksimacija od f *a priori*, nego izvodimo metodu i dojavimo problem ako se pojavi (preciznije, ako dobijemo $x_n \notin I$ ili $f'(x_n) = 0$ u nekom koraku algoritma).

Primjer 1.1.2. Uzmimo funkciju $f: \mathbb{R} \rightarrow \mathbb{R}$ zadanu s $f(x) = x^2 - 2$. Uočimo da je f klase C^∞ pa je specijalno i derivabilna na cijelom \mathbb{R} . Dodatno, njene dvije nultočke su $\pm\sqrt{2}$. Uzmimo početnu aproksimaciju $x_0 = 10$ i izvršimo tri iteracije Newtonove metode (naravno, ako možemo, ali kao što smo već naglasili, o tome ćemo se brinuti „putem”). Ilustracija izvršavanja se može vidjeti na Slici 1.1.

Tablica dobivenih aproksimacija:

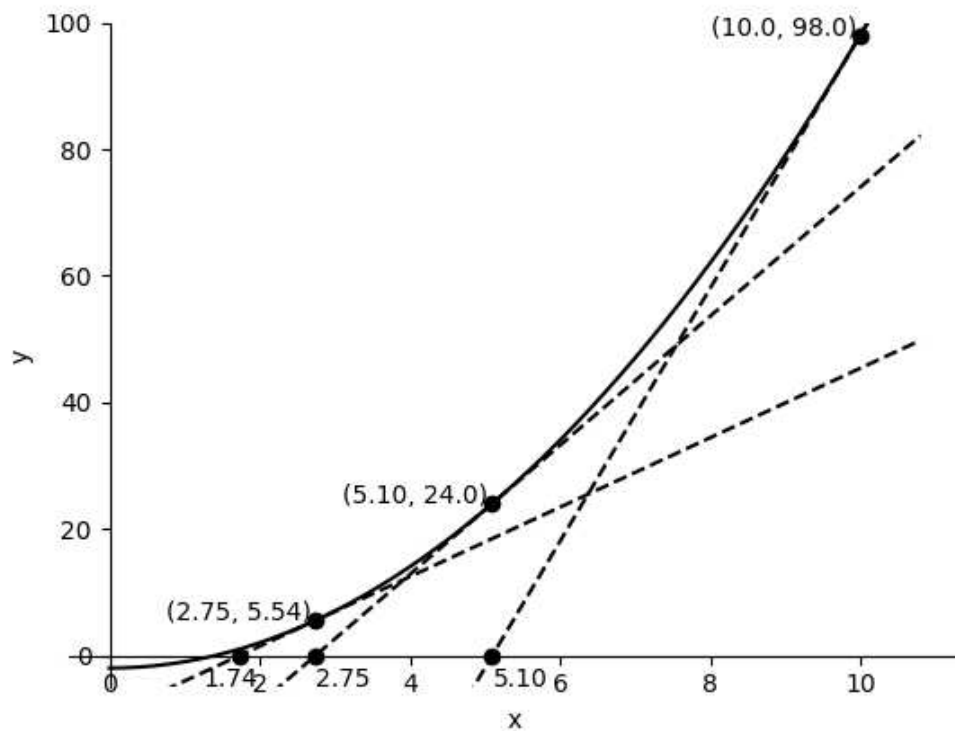
	x_0	x_1	x_2	x_3
Egzaktna vrijednost	10	51/10	2801/1020	9926401/5714040
Zaokružena vrijednost	10.0	5.10	2.75	1.74

Za izračun i Sliku 1.1 smo koristili sljedeći Python kod:

```
import numpy as np
import sympy as sym
import matplotlib.pyplot as plt
from sympy import Float

# Moramo specificirati da je x varijabla.
x = sym.Symbol('x')
# Definiramo funkciju f; ustvari je definiramo
# kao izraz, ne kao funkciju u uzem smislu.
f = x**2 - 2
# Treba nam i derivacija od f.
f_der = sym.diff(f, x)

# Postavljanje plota.
x_plot = np.arange(0.0, 11.0, 0.2)
```



Slika 1.1: Ilustracija izvršavanja Newtonove metode

```
y_plot = np.array([f.subs(x, x_) for x_ in x_plot])
```

```
fig, ax = plt.subplots()
ax.plot(x_plot, y_plot, "k-")
```

```
ax.spines['left'].set_position('zero')
```

```
ax.spines['right'].set_color('none')
ax.yaxis.tick_left()
```

```
ax.spines['bottom'].set_position('zero')
```

```
ax.spines['top'].set_color('none')
ax.xaxis.tick_bottom()

ax.set_xlabel("x")
ax.set_ylabel("y")

ax.set_ylim([-5, 100])

# Numpy array u koji cemo spremati aproksimacije.
x_approx = np.array([10])

for i in range(0, 3):
    if f_der.subs(x, x_approx[i]) == 0:
        raise Exception("Derivacija je jednaka 0.")

    # f(x_n) i f'(x_n)
    f_x = f.subs(x, x_approx[i])
    f_der_x = f_der.subs(x, x_approx[i])

    # Crta tocku (x_n, f(x_n)).
    ax.plot(x_approx[i], f_x, "ko")
    ax.text(x_approx[i] - 2,
            f_x,
            "(" + str(Float(x_approx[i], 3))
            + ", " + str(Float(f_x, 3)) + ")")

    # Crta tangentu na graf od f u tocki (x_n, f(x_n)).
    ax.plot(x_plot,
            f_der_x * x_plot + f_x - f_der_x * x_approx[i],
            "k--")

    # Racuna x_{n+1} i crta ga
    x_novi = x_approx[i] - f_x / f_der_x
    ax.plot(x_novi, 0, "ko")
    x_approx = np.append(x_approx, x_novi)

# Zadnju aproksimaciju malo drugacije crtamo.
if i != 2:
    ax.text(x_novi, -5, str(Float(x_novi, 3)))
```

```

# Crta zadnju aproksimaciju.
ax.text(x_novi - 0.5, -5, str(Float(x_novi, 3)))

# Ispisuje egzaktne vrijednosti aproksimacija.
print(x_approx)

# Racuna i ispisuje zaokružene vrijednosti aproksimacija.
x_approx_f = [Float(x, 3) for x in x_approx]
print(x_approx_f)

# Ovo nam treba da se plot prikaze.
plt.show()

```

1.2 Dokaz konvergencije, ocjena greške, kriterij zaustavljanja za jednostruke nultočke

Pretpostavimo da je f klase C^2 , neka je x_* neka njena nultočka, i $(x_n)_{n \in \mathbb{N}_0}$ niz iteracija dobivenih Newtonovom metodom. Fiksirajmo sada $n \in \mathbb{N}_0$. Po Taylorovom teoremu znamo da postoji ξ_n između x_n i x_* takav da je

$$f(x_*) = f(x_n) + f'(x_n)(x_* - x_n) + \frac{f''(\xi_n)}{2}(x_* - x_n)^2. \quad (1.4)$$

Iskoristimo li da je $f(x_*) = 0$ i pretpostavimo li da je $f'(x_n) \neq 0$, iz jednadžbe (1.4) dobivamo

$$x_* = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f''(\xi_n)}{2f'(x_n)}(x_* - x_n)^2 = x_{n+1} - \frac{f''(\xi_n)}{2f'(x_n)}(x_* - x_n)^2, \quad (1.5)$$

iz čega slijedi i

$$x_* - x_{n+1} = -\frac{f''(\xi_n)}{2f'(x_n)}(x_* - x_n)^2. \quad (1.6)$$

Definicija 1.2.1. Niz iteracija $(x_n)_{n \in \mathbb{N}_0}$ konvergira prema točki x_* s redom konvergencije p , $p \geq 1$, ako vrijedi

$$|x_* - x_n| \leq c|x_* - x_{n-1}|^p, \quad n \in \mathbb{N} \quad (1.7)$$

za neki $c > 0$. Ako je $p = 1$, kažemo da niz konvergira *linearno* prema x_* . Ako je $p = 2$, kažemo da niz konvergira *kvadratično* prema x_* .

Teorem 1.2.2. Neka su f , f' , i f'' neprekidne za sve x u nekom intervalu koji sadrži jedinstvenu nultočku x_* . Ako je početna aproksimacija x_0 izabrana dovoljno blizu nultočke x_* , niz iteracija x_n konvergiraće prema x_* s redom konvergencije $p = 2$ (dakle, kvadratično). Dodatno, vrijedi i

$$\lim_{n \rightarrow \infty} \frac{x_* - x_{n+1}}{(x_* - x_n)^2} = -\frac{f''(x_*)}{2f'(x_*)}. \quad (1.8)$$

Dokaz. Neka je I interval koji sadrži x_* i na kojem su f , f' , i f'' neprekidne. Možemo odabrati $\varepsilon > 0$ takav da je $I_\varepsilon := [x_* - \varepsilon, x_* + \varepsilon] \subseteq I$ te da f' nema nultočku unutar I_ε .

Kako su $|f'|$ i $|f''|$ neprekidne funkcije na I_ε (kao kompozicije neprekidnih funkcija), možemo definirati

$$M := \frac{\max_{x \in I_\varepsilon} |f''(x)|}{2 \min_{x \in I_\varepsilon} |f'(x)|}$$

(jer neprekidna funkcija postiže minimum i maksimum na segmentu). Iz (1.6), za sve $x_0 \in I_\varepsilon$ vrijedi

$$|x_* - x_1| = \frac{|f'(\xi_0)|}{2|f'(x_0)|} (x_* - x_0)^2 \leq \frac{\max_{x \in I_\varepsilon} |f''(x)|}{2 \min_{x \in I_\varepsilon} |f'(x)|} (x_* - x_0)^2 = M(x_* - x_0)^2$$

jer je ξ_0 između x_0 i x_* , dakle, $\xi_0 \in I_\varepsilon$.

Odaberimo sada $x_0 \in I_\varepsilon$ takav da je $M|x_* - x_0| < 1$. Tada vrijedi

$$|x_* - x_1| \leq M(x_* - x_0)^2 < |x_* - x_0| \leq \varepsilon,$$

dakle $x_1 \in I_\varepsilon$. Dodatno,

$$M|x_* - x_1| \leq M^2(x_* - x_0)^2 = (M|x_* - x_0|)^2 < 1.$$

Indukcijom lako pokažemo da vrijedi

$$|x_* - x_n| < \varepsilon, \quad M|x_* - x_n| < 1, \quad n \in \mathbb{N}.$$

Iskoristimo sada opet (1.6) da dobijemo

$$|x_* - x_n| \leq M(x_* - x_{n-1})^2, \quad n \in \mathbb{N}. \quad (1.9)$$

Indukcijom lako dobijemo

$$|x_* - x_n| \leq M^{2^n - 1} (x_* - x_0)^{2^n} \leq \varepsilon (M|x_* - x_0|)^{2^n - 1}, \quad n \in \mathbb{N}.$$

(Koristili smo da je $|x_* - x_n| \leq \varepsilon$.) Kako je $M|x_* - x_0| < 1$, to je $\lim_{n \rightarrow \infty} (M|x_* - x_0|)^{2^n-1} = 0$, dakle

$$\lim_{n \rightarrow \infty} |x_* - x_n| = 0 \iff \lim_{n \rightarrow \infty} x_n = x_*.$$

Drugim riječima, niz iteracija x_n konvergira prema x_* . Činjenica da je konvergencija kvadratična slijedi direktno iz (1.9).

Za zadnju tvrdnju teorema najprije uočimo da $\lim_{n \rightarrow \infty} \xi_n = x_*$ jer je ξ_n između x_n i x_* i $\lim_{n \rightarrow \infty} x_n = x_*$. Korištenjem (1.6) i činjenice da su f' i f'' neprekidne na I_ε dobivamo

$$\lim_{n \rightarrow \infty} \frac{x_* - x_{n+1}}{(x_* - x_n)^2} = - \lim_{n \rightarrow \infty} \frac{f''(\xi_n)}{2f'(x_n)} = - \frac{f''(x_*)}{f'(x_*)}.$$

□

Prethodni teorem pokazuje tzv. *lokalnu* konvergenciju Newtonove metode — ako odaberemo početnu aproksimaciju „dovoljno blizu” jednostruke nultočke x_* , niz iteracija će konvergirati prema x_* . Naravno, jasna je mana tog pristupa; ako ne znamo nultočku funkcije (a prirodno je za pretpostaviti da ne znamo, jer, zašto bismo ju inače išli aproksimirati?) ne znamo jesmo li dovoljno blizu. Srećom, uz još neke dodatne pretpostavke, imamo i *globalnu* konvergenciju Newtonove metode.

Teorem 1.2.3. *Neka je $f \in C^2[a, b]$, $f(a) \cdot f(b) < 0$, i neka f' i f'' nemaju nultočke u $[a, b]$. Neka je $x_0 \in [a, b]$ takva da je $f(x_0) \cdot f''(x_0) > 0$. Tada niz iteracija dobiven Newtonovom metodom, s početnom iteracijom x_0 , konvergira prema (jedinствenoj i jednostrukoj) nultočki od f .*

Dokaz. Kako je f neprekidna i mijenja predznak na $[a, b]$ (što je posljedica uvjeta $f(a) \cdot f(b) < 0$), sigurno ima barem jednu nultočku u $[a, b]$. Dodatno, f' nema nultočke u $[a, b]$ pa ne mijenja predznak (neprekidna je), stoga je f strogo monotona na $[a, b]$ te je nultočka od f jedinstvena. Označimo je s x_* . Zato što f' nema nultočki u $[a, b]$, vrijedi $f'(x_*) \neq 0$, pa je x_* jednostruka nultočka od f .

Uočimo da ni f'' ne mijenja predznak na $[a, b]$. Pretpostavimo sada da vrijedi $f'(x) > 0$ i $f''(x) > 0$ za sve $x \in [a, b]$. Tada je f strogo rastuća funkcija pa mora biti $f(a) < 0$, $f(b) > 0$. Uzmimo $x_0 = b$; naime, znamo da je $f(b) \cdot f''(b) > 0$, i to je trenutno jedina točka za koju znamo da to vrijedi. Neka je $(x_n)_{n \in \mathbb{N}_0}$ niz iteracija generiran Newtonovom metodom; dakle,

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \quad n \in \mathbb{N}.$$

Dokazat ćemo indukcijom da $(x_n)_{n \in \mathbb{N}_0}$ pada i da je omeđen odozdo s x_* , tj. da vrijedi $x_* < x_{n+1} < x_n$ za sve $n \in \mathbb{N}_0$. Za x_0 znamo da vrijedi $x_* < x_0$. Pretpostavimo sada

da za neki $n \in \mathbb{N}_0$ imamo $x_* < x_n$. Tada je $f(x_n) > 0$ jer f strogo raste. Dakle,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} < x_n.$$

Da bismo dokazali $x_{n+1} > x_*$, iskoristimo Taylorov teorem da bismo zapisali

$$f(x_*) = f(x_n) + f'(x_n)(x_* - x_n) + \frac{f''(\xi_n)}{2}(x_* - x_n)^2,$$

gdje je ξ_n između x_* i x_n (znamo i više: $\xi_n \in \langle x_*, x_n \rangle$ jer je $x_n > x_*$). Kako je $f(x_*) = 0$ i $f''(\xi_n) > 0$, lako slijedi

$$f(x_n) + f'(x_n)(x_* - x_n) < 0,$$

iz čega slijedi

$$x_* < x_n - \frac{f(x_n)}{f'(x_n)} = x_{n+1}.$$

(Kratka napomena: sjetimo se da je $f'(x_n) > 0$, što znači da možemo dijeliti s $f'(x_n)$ i pritom ne mijenjamo znak nejednakosti.)

Pokazali smo da je niz $(x_n)_{n \in \mathbb{N}_0}$ uistinu padajući i da je ograničen odozdo s x_* . Možemo zaključiti i da je konvergentan; neka je $\beta = \lim_{n \rightarrow \infty} x_n$. Iz definicijske jednakosti iteracija i neprekidnosti f i f' vrijedi

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} \left[x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \right] \implies \beta = \beta - \frac{f(\beta)}{f'(\beta)} \implies f(\beta) = 0.$$

No, x_* je jedina nultočka od f u $[a, b]$, a mora biti $\beta \in [x_*, x_0] \subseteq [a, b]$, stoga je $\beta = x_*$. Dakle, niz iteracija konvergira prema x_* .

Ostali slučajevi na predznake od f' i f'' se dokazuju posve analogno. \square

Konačno, zanima nas ocjena greške. Pretpostavimo da smo nultočku x_* locirali u nekom segmentu $[a, b]$, neka je f klase C^2 na $[a, b]$ i neka f' nema nultočaka u $[a, b]$. Kao u dokazu Teorema 1.2.3 zaključujemo da je x_* jedinstvena i jednostruka nultočka od f u $[a, b]$. Označimo $m := \min_{x \in [a, b]} |f'(x)|$ i $M := \max_{x \in [a, b]} |f''(x)|$. Uočimo da je $m > 0$. Kao i do sada, niz iteracija označimo s $(x_n)_{n \in \mathbb{N}_0}$ i pretpostavimo da su sve sadržane u $[a, b]$. Iz Taylorovog teorema slijedi

$$f(x_n) = f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) + \frac{f''(\xi_{n-1})}{2}(x_n - x_{n-1})^2,$$

za neki ξ_{n-1} za koji znamo da je između x_n i x_{n-1} , a iz definicijske jednakosti iteracija znamo da vrijedi

$$f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) = 0.$$

Dakle,

$$|f(x_n)| = \frac{|f''(\xi_{n-1})|}{2}(x_n - x_{n-1})^2 \leq \frac{M}{2}(x_n - x_{n-1})^2. \quad (1.10)$$

S druge strane, iz Taylorovog teorema slijedi

$$f(x_n) = f(x_*) + f'(\xi)(x_n - x_*) = f'(\xi)(x_n - x_*),$$

gdje je ξ između x_n i x_* . Dakle,

$$|f(x_n)| = |f'(\xi)||x_n - x_*| \implies |x_n - x_*| = \frac{|f(x_n)|}{|f'(\xi)|} \leq \frac{|f(x_n)|}{m}.$$

Iskoristimo li sada (1.10), dobivamo

$$|x_n - x_*| \leq \frac{M}{2m}(x_n - x_{n-1})^2.$$

Naravno, prethodnu ocjenu greške možemo kombinirati s

$$|x_n - x_*| \leq \frac{|f(x_n)|}{m}.$$

1.3 Newtonova metoda i višestruke nultočke

U prošloj sekciji smo se bavili jednostrukim nultočkama. Pitanje je kako se stvari mijenjaju kada imamo višestruke nultočke. One nam nisu od velikog interesa za ovaj rad pa zato samo navodimo dvije važne propozicije, bez dokaza. Dokaze čitatelj može pronaći u [2].

Propozicija 1.3.1. Neka f ima neprekidnih $p + 1$ derivacija i neka je x_* p -struka nultočka, $p \geq 2$:

$$f(x_*) = f'(x_*) = \dots = f^{(p-1)}(x_*) = 0, \quad f^{(p)}(x_*) \neq 0.$$

Neka je x_0 odabrana dovoljno blizu x_* . Tada niz iteracija $(x_n)_{n \in \mathbb{N}_0}$ konvergira linearno prema x_* .

Uočimo, konvergencija nije više kvadratična, nego je linearna. Pitanje je možemo li to popraviti. Za to nam služi sljedeća propozicija.

Propozicija 1.3.2. Neka je f kao u iskazu Propozicije 1.3.1. Tada možemo pisati $f(x) = (x - x_*)^p h(x)$, $h(x_*) \neq 0$. Definirajmo

$$u: \mathbb{R} \rightarrow \mathbb{R}, \quad u(x) = \frac{f(x)}{f'(x)} = \frac{(x - x_*)h(x)}{ph(x) + (x - x_*)h'(x)}.$$

Tada je x_* jednostruka nultočka od u i ako uzmemo x_0 dovoljno blizu x_* , niz iteracija $(x_n)_{n \in \mathbb{N}_0}$ kojeg dobijemo tako da primijenimo Newtonovu metodu na funkciju u (s početnom iteracijom x_0) konvergira kvadratično prema x_* .

Primjer 1.3.3. Funkcija $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x^4 - 4x^2 + 4$ ima dvije dvostruke nultočke: $\pm\sqrt{2}$. To postaje jasno kada zapišemo $f(x) = (x^2 - 2)^2 = (x - \sqrt{2})^2(x + \sqrt{2})^2$. Dodatno, $f'(x) = 4x(x^2 - 2)$. Sada definirajmo $u: \mathbb{R} \rightarrow \mathbb{R}$ s

$$u(x) = \frac{f(x)}{f'(x)} = \frac{(x^2 - 2)^2}{4x(x^2 - 2)} = \frac{x^2 - 2}{4x}.$$

Ona sada ima dvije jednostruke nultočke: $\pm\sqrt{2}$. No, to su upravo nultočke funkcije f ! Dodatno, iz

$$u'(x) = \frac{x^2 + 2}{4x^2}$$

je jasno da su $\pm\sqrt{2}$ jednostruke nultočke od u . Dakle, po propozicijama iz ove sekcije, više nam se isplati primijeniti Newtonovu metodu na funkciju u nego na funkciju f .

1.4 Implementacija u Pythonu i neki primjeri

Ovu sekciju počinjemo s jednom propozicijom koja će nam biti od koristi pri odabiru kriterija zaustavljanja.

Propozicija 1.4.1. Ako niz $(x_n)_{n \in \mathbb{N}}$ realnih brojeva konvergira prema $x_* \in \mathbb{R}$ kvadratično, tada vrijedi

$$\lim_{n \rightarrow \infty} \frac{|x_{n+1} - x_n|}{|x_n - x_*|} = 1.$$

Dokaz. Zato što je konvergencija kvadratična, postoji $c > 0$ takav da je

$$|x_{n+1} - x_*| \leq c|x_n - x_*|^2, \quad n \in \mathbb{N}_0.$$

Dakle, imamo

$$\left| 1 - \frac{|x_{n+1} - x_n|}{|x_n - x_*|} \right| = \frac{||x_n - x_*| - |x_{n+1} - x_n||}{|x_n - x_*|} \leq \frac{|x_{n+1} - x_*|}{|x_n - x_*|} \leq c|x_n - x_*|,$$

što povlači

$$\lim_{n \rightarrow \infty} \left| 1 - \frac{|x_{n+1} - x_n|}{|x_n - x_*|} \right| = 0 \implies \lim_{n \rightarrow \infty} \frac{|x_{n+1} - x_n|}{|x_n - x_*|} = 1.$$

□

Prethodna propozicija nam daje intuitivno opravdanje zašto umjesto $|x_n - x_*| < \varepsilon$ kao uvjet zaustavljanja možemo gledati $|x_{n+1} - x_n| < \varepsilon$.

Python kod implementacije:

```

import numpy as np
import sympy as sym
import matplotlib.pyplot as plt
from sympy import Float
from sympy.calculus.util import continuous_domain

def glatki_realni_newton(f, var, poc,
                        max_it=1000, m=-1, M=-1,
                        tol=1e-12):
    """ Newtonova metoda za glatku realnu skalarnu funkciju.
    f -- realna funkcija
    var -- varijabla funkcije f (sympy.Symbol objekt)
    poc -- pocetna iteracija
    max_it -- koliko najvise iteracija se izvodi
    m, M -- parametri iz ocjene greske koji se koriste
           za kriterij zaustavljanja
    tol -- tolerancija (default je 1e-12)

    Ukoliko su m i M specificirani, koristi se ocjena
     $|x_n - x_*| \leq M / (2 * m) * (x_n - x_{n-1})^2$ ,
    dakle, ako je  $M / (2 * m) * (x_n - x_{n-1})^2 < tol$ ,
    algoritam se zaustavlja.
    Ukoliko je samo m specificiran, koristi se ocjena
     $|x_n - x_*| \leq |f(x_n)| / m$ ,
    dakle, ako je  $M / (2 * m) * (x_n - x_{n-1})^2 < tol$ ,
    algoritam se zaustavlja, a ako m nije specificiran,
    za kriterij zaustavljanja koristimo
     $|x_{n+1} - x_n| < tol$ .

    Funkcija vraca listu svih aproksimacija.
    """
    f_der = sym.diff(f, var)
    x_approx = np.array([poc])

    for it in range(0, max_it):
        if not continuous_domain(f,
                                var,
                                sym.S.Reals).contains(
                                x_approx[it]):

```

```

        raise Exception("Funkcija nije definirana u "
                        + str(x_approx[it])
                        + "!")
    if not continuous_domain(f_der,
                             var,
                             sym.S.Reals).contains(
                                x_approx[it]):
        raise Exception("Funkcija nije derivabilna u "
                        + str(x_approx[it])
                        + "!")

    if f_der.subs(var, x_approx[it]) == 0:
        raise Exception("Derivacija u "
                        + str(x_approx[it])
                        + " je jednaka 0.")

    f_var = f.subs(var, x_approx[it])
    f_der_var = f_der.subs(var, x_approx[it])
    x_novi = x_approx[it] - f_var/f_der_var
    x_approx = np.append(x_approx, Float(x_novi))

    if m != -1 and M != -1:
        razlika = x_novi - x_approx[it]
        if M / (2 * m) * (razlika)**2 < tol:
            break

    elif m != -1:
        if f.subs(var, x_novi) / m < tol:
            break

    else:
        razlika = x_novi - x_approx[it]
        if np.abs(razlika) < tol:
            break

    return(x_approx)

```

Primjer 1.4.2. Aproximirajmo $\sqrt{2}$. To ćemo napraviti tako da aproksimiramo pozitivnu nultočku funkcije $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = x^2 - 2$. Najprije uočimo da je $f(1) = -1 < 0$ i

$f(2) = 2 > 0$, pa se nultočka od f (za koju znamo da je $\sqrt{2}$) nalazi u segmentu $[1, 2]$. Dodatno, $f'(x) = 2x$ znači da je $f'(x) \neq 0$ na $[1, 2]$ te $m = \min_{x \in [1, 2]} |f'(x)| = 2$. Za drugu derivaciju imamo $f''(x) = 2$ pa je očito $M = \max_{x \in [1, 2]} |f''(x)| = 2$. Primijetimo i da je $f(2) \cdot f''(2) = 4 > 0$. Sada nam Teorem 1.2.3 garantira da će niz iteracija, ako uzmemo $x_0 = 2$, konvergirati prema nultočki od f u $[1, 2]$, tj. $\sqrt{2}$.

Imali smo tri poziva funkcije:

```
x = sym.Symbol("x", Real=True)
proba = glatki_realni_newton(x**2-2, x, Float(2), m=2, M=2)
proba2 = glatki_realni_newton(x**2-2, x, Float(2), m=2)
proba3 = glatki_realni_newton(x**2-2, x, Float(2))
```

Prva dva poziva su završila u 5 iteracija i krajnja aproksimacija je $\sqrt{2} \approx 1.41421356237310$. Treći poziv je napravio jednu iteraciju više, tj. trebalo je 6 iteracija, i krajnja aproksimacija je $\sqrt{2} \approx 1.41421356237309$ (no, nakon „samo” 5 iteracija, aproksimacija je naravno bila $\sqrt{2} \approx 1.41421356237310$, kao i u prva dva poziva).

Primjer 1.4.3. Sljedeći primjer pokazuje da stvarno treba paziti da niz iteracija ne „ispadne” iz domene funkcije f . Uzmimo $f(x) = \ln x$ i probajmo joj naći nultočku s početnom aproksimacijom $x_0 = 3$. Nakon prve iteracije dobijemo $x_1 \approx -0.295836866004330$, što nije u domeni od f .

Primjer 1.4.4. Da Newtonova metoda ne mora nužno konvergirati pokazuje sljedeći primjer. Uzmimo $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = \arctg x$. Njena jedina nultočka je $x = 0$. Probajmo primijeniti Newtonovu metodu s $x_0 = 2$. Nakon poziva

```
x = sym.Symbol("x", Real=True)
proba = glatki_realni_newton(sym.atan(x), x,
                             Float(2), max_it=10)
```

dobijemo sljedeće aproksimacije nultočke:

i	x_i
0	2.000000000000000
1	-3.53574358897045
2	13.9509590869275
3	-279.344066533617
4	122016.998917954
5	-23386004197.9339
6	$8.59076667195035 \times 10^{20}$
7	$-1.15926766989072 \times 10^{42}$
8	$2.11099558761098 \times 10^{84}$
9	$-6.99994339531757 \times 10^{168}$
10	$7.69677752160014 \times 10^{337}$

iz čega je jasno da metoda ne konvergira.

Poglavlje 2

Glatki realni slučaj, poopćenje

2.1 Opis metode i uvodni primjer

Započinjemo slično kao u prvom poglavlju. Za danu $F: O \rightarrow \mathbb{R}^n$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, tražimo $x_* \in O$ takvu da je $F(x_*) = 0$. Pretpostavimo da je F klase C^1 na O . Neka je $x_0 \in O$ aproksimacija nultočke. Promatramo linearnu aproksimaciju od F u točki x_0 ; to je funkcija $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $G(x) = \nabla F(x_0)(x - x_0) + F(x_0)$ (sa $\nabla F(x_0)$ označavamo Jacobijevu matricu od F u točki x_0). Za sljedeću aproksimaciju nultočke od F koristimo nultočku funkcije G , ako G ima nultočku. Drugim riječima, rješavamo jednadžbu

$$\nabla F(x_0)(x - x_0) + F(x_0) = 0.$$

Cilj nam je prvo naći $d_0 \in \mathbb{R}^n$ takav da

$$\nabla F(x_0)d_0 = -F(x_0).$$

Ako takav d_0 postoji (npr. ako je $\nabla F(x_0)$ nesingularna matrica), tada G ima nultočku i ona je dana s

$$x_1 = x_0 + d_0.$$

Ako je $x_1 \in O$ možemo ponoviti postupak: nalazimo $d_1 \in \mathbb{R}^n$ takav da je $\nabla F(x_1)d_1 = -F(x_1)$ (ako takav postoji, naravno) i definiramo $x_2 = x_1 + d_1$. Postupak nastavljamo induktivno i dobivamo sljedeći algoritam.

Algoritam 2.1.1 (Newtonova metoda za glatke sustave). Neka je $F: O \rightarrow \mathbb{R}^n$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, neprekidno diferencijabilna (klase C^1). Neka je $x_0 \in \mathbb{R}^n$ i stavimo $k := 0$.

(1) Ako kriterij zaustavljanja nije zadovoljen, riješi

$$\nabla F(x_k)d_k = -F(x_k)$$

po d_k .

(2) Stavi $x_{k+1} := x_k + d_k$, $k := k + 1$, i vrati se na (1).

Da algoritam bude dobro definiran, mora vrijediti $x_k \in O$ za sve $k \in \mathbb{N}$. Dodatno, rješenje jednadžbe $\nabla F(x_k)d_k = -F(x_k)$ mora postojati za sve $k \in \mathbb{N}_0$. Najbolja situacija je kada je $\nabla F(x_k)$ invertibilna za sve $k \in \mathbb{N}_0$ jer tada imamo osiguranu i egzistenciju i jedinstvenost od d_k (uočimo, ako $\nabla F(x_k)$ nije invertibilna, rješenje od $\nabla F(x_k)d_k = -F(x_k)$ može postojati, ali sigurno nije jedinstveno).

Za kriterij zaustavljanja se može uzeti kriterij oblika

$$\begin{aligned} \|F(x_k)\| &\leq \varepsilon_{\text{rel}}^F \|F(x_0)\| + \varepsilon_{\text{abs}}^F, \\ \|x_{k+1} - x_k\| &\leq \varepsilon_{\text{rel}}^x \|x_0\| + \varepsilon_{\text{abs}}^x, \end{aligned}$$

gdje su $\varepsilon_{\text{rel}}^F$, $\varepsilon_{\text{abs}}^F$, $\varepsilon_{\text{rel}}^x$, i $\varepsilon_{\text{abs}}^x$ neki mali nenegativni realni brojevi (moguće da su neki i 0). Također za kriterij zaustavljanja možemo uzeti i samo jednu od gornje dvije nejednadžbe; recimo, kriterij zaustavljanja može biti

$$\|x_{k+1} - x_k\| \leq 10^{-12}.$$

Primjer 2.1.2. Odredimo rješenje sustava nelinearnih jednadžbi

$$\begin{aligned} 5ye^x - 5e^{5x} &= 0 \\ 5e^x - 5y^4 &= 0. \end{aligned}$$

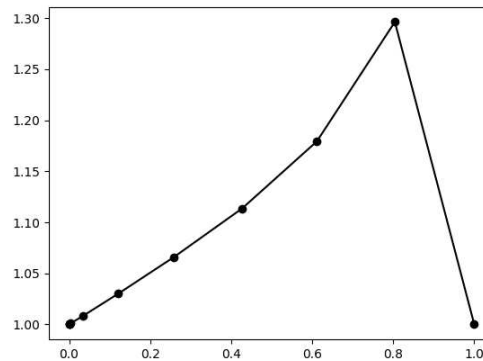
Lako se pokaže da je $(0, 1)$ jedino rješenje sustava. Nas zanima primjena Newtonove metode. Definiramo preslikavanje

$$F: \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad F(x, y) = (5ye^x - 5e^{5x}, 5e^x - 5y^4),$$

za koje se lako vidi da je neprekidno diferencijabilno na cijelom \mathbb{R}^2 te da je Jacobijeva matrica dana s

$$\nabla F(x, y) = \begin{bmatrix} 5ye^x - 25e^{5x} & 5e^x \\ 5e^x & -20y^3 \end{bmatrix}.$$

Za početnu aproksimaciju smo uzeli $(1, 1)$, izvrtili 10 iteracija, i dobili smo točku jako blizu $(0, 1)$, kako smo i očekivali. Aproksimacije i njihovo „kretanje” možemo vidjeti na Slici 2.1.



Slika 2.1: Kretanje aproksimacija u Primjeru 2.1.2, početna aproksimacija: (1, 1)

Python kod:

```

import numpy as np
import sympy as sym
import matplotlib.pyplot as plt
from sympy.functions.elementary.exponential import exp

# Varijable.
x = sym.Symbol("x")
y = sym.Symbol("y")

# Funkcija F
F = sym.Matrix([5*y*exp(x) - 5*exp(5*x), 5*exp(x) - 5*y**4])

# Jacobijeva matrica funkcije F.
dF = F.jacobian(sym.Matrix([x, y]))

# Ovdje spremamo aproksimacije.
# Zasad je spremljena pocetna aproksimacija.
approx = np.array([1, 1], dtype=float)

# Ovdje spremamo aproksimacije u formatu za plotanje.
approx_plot_x = [approx[0]]
approx_plot_y = [approx[1]]

# 10 iteracija

```



```
for it in range(0, 10):
    # Izracunaj  $f(x_k, y_k)$ . Treba prebaciti u Numpy Matrix.
    f = F.subs([(x, approx[0]), (y, approx[1])])
    f = np.matrix(f, dtype=float)

    # Slicno i za Jacobijevu matricu.
    df = dF.subs([(x, approx[0]), (y, approx[1])])
    df = np.matrix(df, dtype=float)

    # Rijeci nabra  $F(x_k, y_k)$   $d_k = -F(x_k, y_k)$ .
    d = np.linalg.solve(df, -f)
    # Prebaci  $d$  u odgovarajuci format.
    d = np.squeeze(np.asarray(d))

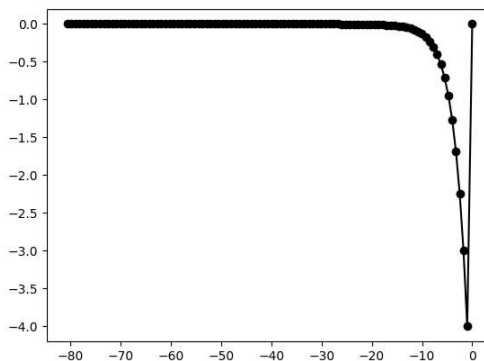
    #  $(x_{k+1}, y_{k+1}) = (x_k, y_k) + d_k$ 
    approx = np.add(approx, d)

    approx_plot_x.append(approx[0])
    approx_plot_y.append(approx[1])

plt.plot(approx_plot_x, approx_plot_y, "o-k")

plt.show()
```

Što ako za početnu aproksimaciju uzmemo $(0, 0)$? Nakon 100 iteracija smo se poprilično udaljili od točke $(0, 1)$ (rješenja sustava), kao što se vidi na Slici 2.2. Ako povećamo broj iteracija, još više se udaljimo od pravog rješenja sustava. U ovom slučaju smo poprilično sigurni da metoda **ne konvergira**.

Slika 2.2: Kretanje aproksimacija u Primjeru 2.1.2, početna aproksimacija: $(0, 0)$

2.2 Lokalna konvergencija Newtonove metode za glatke sustave

U ovoj sekciji ćemo se baviti pitanjem lokalne konvergencije Newtonove metode za glatke sustave. Drugim riječima, naći ćemo dovoljne uvjete pod kojima smo sigurni da metoda konvergira dok god za početnu aproksimaciju uzmemo neku točku koja je „dovoljno blizu” nultočki.

Smatramo da nam je unaprijed zadana neka vektorska norma $\|\cdot\|$. Sa $K(x, r)$ označavamo otvorenu kuglu oko x radijusa r ; drugim riječima,

$$K(x, r) = \{y \in \mathbb{R}^n \mid \|x - y\| < r\}.$$

Umjesto $(\nabla F(x))^{-1}$ pisat ćemo jednostavno $\nabla F(x)^{-1}$. Za matričnu normu ćemo promatrati operatorsku normu induciranu danom vektorskom normom $\|\cdot\|$; drugim riječima, za $A \in \mathbb{R}^{n \times n}$ imamo

$$\|A\| = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|}{\|x\|}.$$

Ta norma je *konzistentna*; za sve $A, B \in \mathbb{R}^{n \times n}$ vrijedi

$$\|AB\| \leq \|A\| \|B\|.$$

Dodatno, za sve $A \in \mathbb{R}^{n \times n}$ i $x \in \mathbb{R}^n$ vrijedi

$$\|Ax\| \leq \|A\| \|x\|.$$

Za jediničnu matricu I imamo $\|I\| = 1$.

Prije nego se pozabavimo lokalnom konvergencijom Newtonove metode, učinit ćemo malu digresiju da bismo dokazali jednu važnu lemu vezanu uz operatorske norme.

Lema 2.2.1. *Neka je $\|\cdot\|$ neka operatorska norma na $\mathbb{R}^{n \times n}$ inducirana vektorskom normom $\|\cdot\|$. Neka je $E \in \mathbb{R}^{n \times n}$ takva da je $\|E\| < 1$. Tada je $I - E$ invertibilna matrica i vrijedi*

$$\|(I - E)^{-1}\| \leq \frac{1}{1 - \|E\|}.$$

Ako je $A \in \mathbb{R}^{n \times n}$ invertibilna i $B \in \mathbb{R}^{n \times n}$ takva da je $\|A^{-1}(A - B)\| < 1$, tada je B invertibilna i

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}(A - B)\|}.$$

Dokaz. Pretpostavimo da $I - E$ nije invertibilna. Tada postoji neki $y \in \mathbb{R}^n$, $y \neq 0$ takav da je $(I - E)y = 0$, tj. $Ey = y$. Sada imamo

$$\|E\| = \max_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ex\|}{\|x\|} \geq \frac{\|Ey\|}{\|y\|} = 1,$$

što je kontradikcija s pretpostavkom da je $\|E\| < 1$. Dakle, $I - E$ mora biti invertibilna.

Dodatno,

$$\begin{aligned} 1 = \|I\| &= \|(I - E)(I - E)^{-1}\| = \|(I - E)^{-1} - E(I - E)^{-1}\| \\ &\geq \|(I - E)^{-1}\| - \|E(I - E)^{-1}\| \geq \|(I - E)^{-1}\| - \|E\| \|(I - E)^{-1}\| \\ &= (1 - \|E\|)\|(I - E)^{-1}\|, \end{aligned}$$

iz čega slijedi

$$\|(I - E)^{-1}\| \leq \frac{1}{1 - \|E\|}, \quad (2.1)$$

što se i tražilo.

Dokažimo sada i drugu tvrdnju leme. Iz prve tvrdnje i pretpostavke $\|A^{-1}(A - B)\| < 1$ znamo da je $I - A^{-1}(A - B)$ invertibilna. Kako je

$$I - A^{-1}(A - B) = I - A^{-1}A + A^{-1}B = A^{-1}B$$

invertibilna, B je invertibilna kao produkt dviju invertibilnih matrica: $B = A(A^{-1}B)$.

Stavimo li $E = A^{-1}(A - B)$, tada je

$$(I - E)^{-1} = (A^{-1}B)^{-1} = B^{-1}A.$$

Konačno, koristeći (2.1) sada možemo dokazati zadnji dio leme:

$$\|B^{-1}\| = \|B^{-1}AA^{-1}\| \leq \|A^{-1}\| \|B^{-1}A\| = \|A^{-1}\| \|(I - E)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|E\|} = \frac{\|A^{-1}\|}{1 - \|A^{-1}(A - B)\|}.$$

□

Prethodna lema nam omogućuje da dokažemo sljedeću lemu, koja će nam biti od direktne koristi pri dokazivanju lokalne konvergencije.

Lema 2.2.2. *Neka je $F: O \rightarrow \mathbb{R}^n$, gdje je $O \subseteq \mathbb{R}^n$ otvoren skup, neprekidno diferencijabilna i neka je ∇F Lipschitz-neprekidna na O s konstantom $L > 0$. Neka je $z \in O$ takav da je $\nabla F(z)$ invertibilna te neka je $\beta > 0$ takav da je $\|\nabla F(z)^{-1}\| \leq \beta$. Tada za sve $x \in K(z, r)$, gdje je $0 < r < \frac{c}{L\beta}$ i $0 < c < 1$ fiksna, $\nabla F(x)$ invertibilna i vrijedi*

$$\|\nabla F(x)^{-1}\| \leq \frac{\beta}{1-c}.$$

Dokaz. Neka je $x \in K(z, r)$ proizvoljan. Tada je

$$\begin{aligned} \|\nabla F(z)^{-1}(\nabla F(z) - \nabla F(x))\| &\leq \|\nabla F(z)^{-1}\| \|\nabla F(z) - \nabla F(x)\| \leq \beta \cdot L\|z - x\| \\ &< \beta Lr < \beta L \cdot \frac{c}{L\beta} = c < 1 \end{aligned}$$

(u drugoj nejednakosti smo iskoristili da je ∇F Lipschitz-neprekidna s konstantom L). Po Lemi 2.2.1 (točnije, iz njene druge tvrdnje) sada znamo da je $\nabla F(x)$ invertibilna te da vrijedi

$$\|\nabla F(x)^{-1}\| \leq \frac{\|\nabla F(z)^{-1}\|}{1 - \|\nabla F(z)^{-1}(\nabla F(z) - \nabla F(x))\|} \leq \frac{\beta}{1-c}.$$

□

Konačno, na red je došao teorem lokalne konvergencije.

Teorem 2.2.3. *Neka je $F: O \rightarrow \mathbb{R}^n$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, neprekidno diferencijabilna. Neka postoje $x_* \in \mathbb{R}^n$, $r > 0$, i $\beta > 0$ takvi da je $F(x_*) = 0$, $\nabla F(x_*)$ je invertibilna, $\|\nabla F(x_*)^{-1}\| \leq \beta$, i ∇F Lipschitz-neprekidna (s konstantom $L > 0$) na $K(x_*, r) \subseteq O$. Tada postoji $\varepsilon > 0$ takav da za sve $x_0 \in K(x_*, \varepsilon)$ niz iteracija $(x_n)_{n \in \mathbb{N}_0}$ dobiven Newtonovom metodom (Algoritmom 2.1.1) je dobro definiran, konvergira prema x_* , i zadovoljava*

$$\|x_{k+1} - x_*\| \leq \frac{L\beta}{2(1-c)} \|x_k - x_*\|^2, \quad k \in \mathbb{N}_0,$$

za neki fiksni $0 < c \leq \frac{2}{3}$.

Dokaz. Fiksirajmo $0 < c \leq \frac{2}{3}$ i definirajmo

$$\varepsilon := \min \left\{ r, \frac{c}{L\beta} \right\}.$$

Primjenom Leme 2.2.2 ($z := x_*$ i $x := x_0$) slijedi da je $\nabla F(x_0)$ invertibilna i da vrijedi

$$\|\nabla F(x_0)^{-1}\| \leq \frac{\beta}{1-c}.$$

Kako je $\nabla F(x_0)$ invertibilna, x_1 je dobro definiran i vrijedi

$$\begin{aligned}
 x_1 - x_* &= (x_0 + d_0) - x_* \\
 &= (x_0 - \nabla F(x_0)^{-1}F(x_0)) - x_* \\
 &= x_0 - x_* - \nabla F(x_0)^{-1}F(x_0) \\
 &= x_0 - x_* - \nabla F(x_0)^{-1}(F(x_0) - F(x_*)) \\
 &= \nabla F(x_0)^{-1}(F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0)).
 \end{aligned}$$

Dakle,

$$\begin{aligned}
 \|x_1 - x_*\| &= \|\nabla F(x_0)^{-1}(F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0))\| \\
 &\leq \|\nabla F(x_0)^{-1}\| \|F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0)\| \\
 &\leq \frac{\beta}{1 - c} \|F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0)\|.
 \end{aligned}$$

Stavimo li $d := x_* - x_0$, dobivamo

$$F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0) = F(x_0 + d) - F(x_0) - \nabla F(x_0)d.$$

Po teoremu srednje vrijednosti (integralni oblik) dobivamo

$$F(x_0 + d) - F(x_0) = \int_0^1 \nabla F(x_0 + td)d dt,$$

pa je

$$\begin{aligned}
 \|F(x_0 + d) - F(x_0) - \nabla F(x_0)d\| &= \left\| \int_0^1 \nabla F(x_0 + td)d dt - \nabla F(x_0)d \right\| \\
 &= \left\| \int_0^1 (\nabla F(x_0 + td)d - \nabla F(x_0)d) dt \right\| \\
 &\leq \int_0^1 \|\nabla F(x_0 + td)d - \nabla F(x_0)d\| dt \\
 &\leq \int_0^1 \|\nabla F(x_0 + td) - \nabla F(x_0)\| \|d\| dt \\
 &\leq \|d\| \int_0^1 L\|x_0 + td - x_0\| dt \\
 &= L\|d\| \int_0^1 \|td\| dt \\
 &= \frac{L\|d\|^2}{2},
 \end{aligned}$$

gdje smo u trećem redu nejednakost trokuta za integrale, u petom redu smo koristili da je ∇F Lipschitz-neprekidna s konstantom L na $K(x_*, r)$, a zadnja jednakost vrijedi jer je

$$\int_0^1 \|td\| dt = \int_0^1 t\|d\| dt = \|d\| \int_0^1 t dt = \frac{\|d\|}{2}.$$

Dakle,

$$\|F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0)\| \leq \frac{L\|x_* - x_0\|^2}{2} = \frac{L\|x_0 - x_*\|^2}{2},$$

te je

$$\begin{aligned} \|x_1 - x_*\| &\leq \frac{\beta}{1-c} \|F(x_*) - F(x_0) - \nabla F(x_0)(x_* - x_0)\| \\ &\leq \frac{\beta}{1-c} \cdot \frac{L\|x_0 - x_*\|^2}{2} \\ &= \frac{L\beta\|x_0 - x_*\|^2}{2(1-c)}. \end{aligned}$$

Pokažimo da je dodatno $\|x_1 - x_*\| < \varepsilon$. Iskoristimo da je $\|x_0 - x_*\| < \varepsilon$ i $\varepsilon \leq \frac{c}{L\beta}$:

$$\|x_1 - x_*\| \leq \frac{L\beta\|x_0 - x_*\|^2}{2(1-c)} < \frac{L\beta\varepsilon}{2(1-c)}\|x_0 - x_*\| \leq \frac{c}{2(1-c)}\|x_0 - x_*\| \leq \|x_0 - x_*\| < \varepsilon.$$

Koristili smo da je

$$\frac{c}{2(1-c)} \leq \frac{\frac{2}{3}}{\frac{1}{3} \cdot 2} = 1$$

jer je $c \leq \frac{2}{3}$.

Dokazali smo da je $x_1 \in K(x_*, \varepsilon)$ te da vrijedi

$$\|x_1 - x_*\| \leq \frac{L\beta}{2(1-c)}\|x_0 - x_*\|^2.$$

Analogno, indukcijom se pokaže da za sve $k \in \mathbb{N}_0$ imamo da je $x_k \in K(x_*, \varepsilon)$ i

$$\|x_{k+1} - x_*\| \leq \frac{L\beta}{2(1-c)}\|x_k - x_*\|^2. \quad (2.2)$$

Prethodna nejednakost pokazuje i da je $\|x_{k+1} - x_*\| < \|x_k - x_*\|$, jer imamo

$$\|x_{k+1} - x_*\| \leq \frac{L\beta}{2(1-c)}\|x_k - x_*\|^2 < \frac{L\beta\varepsilon}{2(1-c)}\|x_k - x_*\| \leq \frac{c}{2(1-c)}\|x_k - x_*\| \leq \|x_k - x_*\|.$$

Promatrajmo sada niz realnih brojeva $(\alpha_k)_{k \in \mathbb{N}_0}$, $\alpha_k := \|x_k - x_*\|$. Taj niz je omeđen odozdo s 0 i padajuć je. Zaključujemo da postoji $\alpha := \lim_{k \rightarrow \infty} \alpha_k$. Iz nejednadžbe (2.2) dobivamo

$$\|x_{k+1} - x_*\| < \frac{L\beta\varepsilon}{2(1-c)}\|x_k - x_*\| \leq \frac{c}{2(1-c)}\|x_k - x_*\|.$$

Puštanjem limesa $k \rightarrow \infty$ dobivamo

$$\alpha \leq \frac{c}{2(1-c)}\alpha.$$

Kako je $\alpha_k \geq 0$ za sve $k \in \mathbb{N}_0$, znamo da je $\alpha \geq 0$. Pretpostavimo li da je $\alpha > 0$ slijedi

$$1 \leq \frac{c}{2(1-c)}.$$

Drugim riječima, ako uzmemo dovoljno mali c osigurat ćemo da je $\alpha = 0$, tj. da je

$$\lim_{k \rightarrow \infty} x_k = x_*.$$

□

Završit ćemo ovu sekciju s raspravom vezanom uz brzinu konvergencije Newtonove metode za glatke sustave.

Definicija 2.2.4. Niz iteracija $(x_n)_{n \in \mathbb{N}_0}$ konvergira prema točki x_* s redom konvergencije p , $p \geq 1$, ako vrijedi

$$\|x_* - x_n\| \leq c\|x_* - x_{n-1}\|^p, \quad n \in \mathbb{N}$$

za neki $c > 0$. Ako je $p = 1$, kažemo da niz konvergira *linearno* prema x_* . Ako je $p = 2$, kažemo da niz konvergira *kvadratično* prema x_* .

Odmah vidimo da teorem lokalne konvergencije Newtonove metode povlači da je konvergencija kvadratična. Dodatno,

$$\left| 1 - \frac{\|x_{k+1} - x_k\|}{\|x_k - x_*\|} \right| = \frac{\left| \|x_k - x_*\| - \|x_{k+1} - x_k\| \right|}{\|x_k - x_*\|} \leq \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \leq \frac{L\beta}{2(1-c)}\|x_k - x_*\|,$$

što povlači

$$\lim_{k \rightarrow \infty} \left| 1 - \frac{\|x_{k+1} - x_k\|}{\|x_k - x_*\|} \right| = 0 \implies \lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_k\|}{\|x_k - x_*\|} = 1.$$

Taj limes nam daje intuitivno opravdanje zašto umjesto $\|x_k - x_*\| < \varepsilon$ kao uvjet zaustavljanja možemo gledati $\|x_{k+1} - x_k\| < \varepsilon$.

2.3 Globalna konvergencija

Globalnom konvergencijom se u ovome radu nećemo detaljnije baviti, ali ćemo ipak spomenuti neke rezultate i navesti jednu strategiju kako postići globalnu konvergenciju.

Teorem 2.3.1 (Kantorovič). *Neka je $r > 0$, $x_0 \in \mathbb{R}^n$, $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ i pretpostavimo da je F neprekidno diferencijabilna na $K(x_0, r)$. Pretpostavimo da je ∇F Lipschitz-neprekidna na $K(x_0, r)$ s konstantom L i neka je $\nabla F(x_0)$ invertibilna. Dodatno, pretpostavimo da postoje konstante β, η takve da*

$$\|\nabla F(x_0)^{-1}\| \leq \beta, \quad \|\nabla F(x_0)^{-1}F(x_0)\| \leq \eta.$$

Definiramo $\gamma_r := \beta L$ i $\alpha := \gamma_r \eta$. Ako je $\alpha \leq \frac{1}{2}$ i $r \geq r_0 = (1 - \sqrt{1 - 2\alpha})/\gamma_r$, tada je niz iteracija $(x_k)_{k \in \mathbb{N}_0}$ dobiven primjenom Algoritma 2.1.1 dobro definiran i konvergira prema x_* , jedinstvenoj nultočki od F u zatvorenju od $K(x_0, r_0)$. Ako je $\alpha < \frac{1}{2}$, tada je x_* jedinstvena nultočka od F u $K(x_0, r_1)$, gdje je $r_1 = \min\{r, (1 + \sqrt{1 - 2\alpha})/\gamma_r\}$ i vrijedi

$$\|x_k - x_*\| \leq (2\alpha)^{2^k} \frac{\eta}{\alpha}, \quad k \in \mathbb{N}_0.$$

Dokaz prethodnog teorema se može pronaći u [3].

Da bismo osigurali globalnu konvergenciju metode, često se korak (2) Algoritma 2.1.1 zamjenjuje s

$$x_{k+1} = x_k + \alpha_k d_k,$$

gdje je $(\alpha_k)_{k \in \mathbb{N}_0}$ niz brojeva takvih da $\alpha_k \in \langle 0, 1 \rangle$; obično se naziva *parametar prigušenja*.

Postoji mnogo različitih strategija izbora α_k . Jedna je da uzmemo $\alpha_k = \omega^l$, gdje je $\omega \in \langle 0, 1 \rangle$ fiksiran, a $l \in \mathbb{N}_0$ najmanji takav da vrijedi

$$\|F(x_k + \omega^l d_k)\| \leq (1 - \nu \omega^l) \|F(x_k)\|, \quad (2.3)$$

gdje je $\nu \in \langle 0, 1 \rangle$ fiksirani parametar. Uvjet (2.3) se naziva *uvjet dovoljnog smanjenja*. Ova upravo opisana realizacija se naziva *Armijovo pravilo veličine koraka*.

Napomenimo da uvjet oblika

$$\|F(x_k + \alpha_k d_k)\| < \|F(x_k)\|$$

ne može zamijeniti uvjet (2.3) jer niz iteracija $(x_k)_{k \in \mathbb{N}_0}$ može „zapeti” u nekom \bar{x} koji nije nultočka od F te stoga ne možemo dokazati konvergenciju.

2.4 Newtonova metoda i bezuvjetna optimizacija

U ovoj sekciji ćemo opisati jednu čestu uporabu Newtonove metode za glatke sustave. Neka je $f: O \rightarrow \mathbb{R}$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, neka funkcija. Cilj nam je pronaći (naravno, ako postoje) lokalne minimume/maksimume funkcije f . Sjetimo se, $x_* \in O$ je (*strogi*) *lokalni minimum* od f ako postoji $r > 0$ takav da je $f(x_*) \leq f(x)$ ($f(x_*) < f(x)$) za sve $x \in O \cap (K(x_*, r) \setminus \{x_*\})$. Analogno, $x_* \in O$ je (*strogi*) *lokalni maksimum* od f ako postoji $r > 0$ takav da je $f(x_*) \geq f(x)$ ($f(x_*) > f(x)$) za sve $x \in O \cap (K(x_*, r) \setminus \{x_*\})$.

Od velike pomoći pri ovoj potrazi su nam sljedeća dva teorema koje navodimo bez dokaza jer smatramo da su to općepoznati teoremi.

Teorem 2.4.1 (Fermatova lema). *Neka je $f: O \rightarrow \mathbb{R}$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, diferencijabilna funkcija. Ako je x_* lokalni minimum ili lokalni maksimum funkcije f , tada vrijedi $\nabla f(x_*) = 0$.*

Za funkciju $f: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, s $H_f(x)$ označavamo Hesseovu matricu funkcije f u točki $x \in O$.

Teorem 2.4.2. *Neka je $f: O \rightarrow \mathbb{R}$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, klase C^2 . Neka je $x_* \in O$ takav da $\nabla f(x_*) = 0$.*

Ako je $H_f(x_)$ pozitivno definitna matrica, onda je x_* lokalni minimum od f . Ako je x_* lokalni maksimum od f , onda je $H_f(x_*)$ negativno definitna matrica.*

Ako je $H_f(x_)$ pozitivno semidefinitna matrica, onda je x_* lokalni minimum od f . Ako je x_* lokalni maksimum od f , onda je $H_f(x_*)$ negativno semidefinitna matrica.*

Opišimo sada jedno moguće rješenje problema bezuvjetne optimizacije. Neka je $f: O \rightarrow \mathbb{R}$, gdje je $O \subseteq \mathbb{R}^n$ otvoren, klase C^2 . Da bismo našli lokalne ekstreme funkcije f , najprije tražimo nultočke od ∇f . Definiramo li $F: O \rightarrow \mathbb{R}^n$, $F(x) = \nabla f(x)$, možemo iskoristiti Newtonovu metodu za glatke sustave, jer znamo da je $\nabla F = H_f$.

Primjer 2.4.3. Promatrajmo $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x, y) = 5ye^x - e^{5x} - y^5$. Njena Jacobijeva matrica je

$$\nabla f(x, y) = \begin{bmatrix} 5ye^x - 5e^{5x} & 5e^x - 5y^4 \end{bmatrix}.$$

Dakle, definiramo $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $F(x, y) = (5ye^x - 5e^{5x}, 5e^x - 5y^4)$.

Primijetimo sada da je to ista funkcija kao u Primjeru 2.1.2! Prisjetimo se da ako za početnu aproksimaciju uzmemo $(1, 1)$, niz iteracija dobiven Newtonovom metodom se približava točki $(0, 1)$. Dodatno, ručnom provjerom vidimo da je $F(0, 1) = (0, 0)$. Dakle, $(0, 1)$ je nultočka od ∇f , pa je to kandidat za lokalni ekstrem.

Dodatno,

$$H_f(x, y) = \nabla F(x, y) = \begin{bmatrix} 5ye^x - 25e^{5x} & 5e^x \\ 5e^x & -20y^3 \end{bmatrix},$$

te je

$$H_f(0, 1) = \begin{bmatrix} -20 & 5 \\ 5 & -20 \end{bmatrix}.$$

Analizom te matrice dobijemo da je ona negativno definitna (npr. možemo iskoristiti *Sylvesterov kriterij*). Dakle, $(0, 1)$ je lokalni maksimum od f .

2.5 Implementacija u Pythonu

Implementacija je dosta slična kao u skalarnom smislu, s tim da Jacobijevom matricom baratamo kao u Primjeru 2.1.2.

```
import numpy as np
import sympy as sym
import matplotlib.pyplot as plt

def glatki_vektorski_newton(F, var, poc,
                           max_it=1000, tol=1e-12):
    """ Newtonova metoda za glatku realnu skalarnu funkciju.
    F -- vektorska funkcija
    var -- sympy.Matrix objekt koji sadrzi
           varijable funkcije F (sadrzi sympy.Symbol objekte)
    poc -- pocetna iteracija
    max_it -- koliko najviše iteracija se izvodi
    tol -- tolerancija

    Za kriterij zaustavljanja koristimo
    ||x_{n+1} - x_n|| < tol.

    Funkcija vraca listu svih aproksimacija.
    """
    dF = F.jacobian(var)
    x_approx = np.array([poc])

    for it in range(0, max_it):
        # Ovdje valja istaknuti da nazalost
        # izgleda da jos uvijek ne postoji
        # neki analogon continuous_domain funkcije.

        # Takoder, sto se tice (ne)invertibilnosti,
```

```

# pustit cemo Pythonu da procijeni pri
# rjesavanju sustava.

vrijednost_varijabli = [(var[i], x_approx[it][i])
                        for i in range(0, len(var))]
F_var = F.subs(vrijednost_varijabli)
F_var = np.matrix(F_var, dtype=float)

dF_var = dF.subs(vrijednost_varijabli)
dF_var = np.matrix(dF_var, dtype=float)

d = np.linalg.solve(dF_var, -F_var)
d = np.squeeze(np.asarray(d))

x_novi = np.add(x_approx[it], d)
x_approx = np.vstack([x_approx, x_novi])

if np.linalg.norm(d) < tol:
    break
return(x_approx)

```

Primjer 2.5.1. Funkcija $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $F(x, y) = (x^2 - y^2, -2xy)$ ima jedinstvenu nultočku $(0, 0)$. Aproksimirajmo tu nultočku Newtonovom metodom s početnom aproksimacijom $(1, 1)$:

```

x = sym.Symbol("x", Real=True)
y = sym.Symbol("y", Real=True)

F = sym.Matrix([x**2 - y**2, -2*x*y])
proba = glatki_vektorski_newton(F, sym.Matrix([x, y]),
                               np.array([1, 1]))

```

Nakon 41 iteracije dobijemo $(4.54747351 \times 10^{-13}, 4.54747351 \times 10^{-13})$, što je skoro $(0, 0)$.

Poglavlje 3

Generalizacija Newtonove metode

3.1 Uvod, generalizirani diferencijal i generalizirana Newtonova metoda

Često nelinearno preslikavanje $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ nije nužno diferencijabilno u klasičnom (Fréchetovom) smislu. Ipak, htjeli bismo razviti neku metodu da nađemo $x_* \in \mathbb{R}^n$ takve da je $F(x_*) = 0$, tj. nultočke od F . U ovom poglavlju nam je cilj poopćiti Newtonovu metodu za glatke sustave i dobiti metodu koja je upotrebljiva za širi skup sustava.

Općenito, baviti ćemo se funkcijama koje su *lokalno Lipschitz-neprekidne* jer za njih vrijedi sljedeći važan rezultat.

Teorem 3.1.1 (Rademacher). *Ako je $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ lokalno Lipschitz-neprekidna, onda je gotovo svuda diferencijabilna.*

Nadalje sa D_F označavamo podskup od \mathbb{R}^n na kojem je F diferencijabilna. Cilj nam je uvesti „nešto” što će zamijeniti diferencijal u točkama skupa $\mathbb{R}^n \setminus D_F$. Zato uvodimo B-subdiferencijale („B” dolazi od „Bouligand”, koji ih je uveo) i generalizirane diferencijale.

Definicija 3.1.2. Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$, gdje je O otvoren, lokalno Lipschitz-neprekidna u $x \in O$.

(1) Skup

$$\partial_B F(x) := \{G \in \mathbb{R}^{n \times m} \mid \exists (x_k) \subseteq D_F \text{ takav da } x_k \rightarrow x, \nabla F(x_k) \rightarrow G\}$$

nazivamo *B-subdiferencijal* od F u x .

(2) Za skup S , sa $\text{co}(S)$ označimo njegovu konveksnu ljusku. Skup

$$\partial F(x) := \text{co}(\partial_B F(x))$$

nazivamo *generalizirani diferencijal* od F u x . Često koristimo i naziv *Clarkeov generalizirani diferencijal*. U slučaju $m = 1$, $\partial F(x)$ se često naziva i *generalizirani gradijent*.

Primjer 3.1.3. Promotrimo $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = |x|$. Ona je očito lokalno Lipschitz-neprekidna te je derivabilna za sve $x \neq 0$:

$$f'(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \end{cases}.$$

Dodatno, f nije derivabilna u 0. Dakle, $D_f = \mathbb{R} \setminus \{0\}$.

U slučaju funkcija s \mathbb{R} na \mathbb{R} znamo da možemo poistovijetiti ∇f s f' (jer $\mathbb{R}^{1 \times 1}$ poistovjećujemo s \mathbb{R}). Nađimo sada $\partial_B f(0)$. Za niz $(x_n)_{n \in \mathbb{N}}$, $x_n = \frac{1}{n}$, imamo $x_n \rightarrow 0$ i $f'(x_n) = 1 \rightarrow 1$. Dakle, $1 \in \partial_B f(0)$. Za niz $(y_n)_{n \in \mathbb{N}}$, $y_n = -\frac{1}{n}$, imamo $y_n \rightarrow 0$ i $f'(y_n) = -1 \rightarrow -1$. Dakle, $-1 \in \partial_B f(0)$. Iz ovog zaključujemo $\{-1, 1\} \subseteq \partial_B f(0)$.

Neka je sada $\alpha \in \partial_B f(0)$. To znači da postoji niz $(x_n) \subseteq \mathbb{R} \setminus \{0\}$ (jer je $D_f = \mathbb{R} \setminus \{0\}$) takav da $x_n \rightarrow 0$ i $f'(x_n) \rightarrow \alpha$. No, kako je $f'(x_n) = \pm 1$ za sve $n \in \mathbb{N}$, da bi niz $(f'(x_n))_{n \in \mathbb{N}}$ konvergirao mora postojati $n_0 \in \mathbb{N}$ takav da je taj niz konstantan nakon n_0 — drugim riječima, $f'(x_m) = f'(x_{n_0})$ za sve $m \geq n_0$. Kako je $f'(x_{n_0}) = \pm 1$, zaključujemo $\alpha = \pm 1$. Dakle, $\partial_B f(0) \subseteq \{-1, 1\}$, te je onda $\partial_B f(0) = \{-1, 1\}$.

Generalizirani diferencijal of f u 0, $\partial f(0)$, je konveksna ljuska od $\partial_B f(0)$:

$$\partial f(0) = \{\lambda \cdot (-1) + (1 - \lambda) \cdot 1 \mid \lambda \in [0, 1]\} = \{1 - 2\lambda \mid \lambda \in [0, 1]\} = [-1, 1].$$

Razlog zašto B-subdiferencijal i generalizirani diferencijal smatramo poopćenjima klasičnog diferencijala je opisan u sljedećoj propoziciji.

Propozicija 3.1.4. Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$, gdje je O otvoren, lokalno Lipschitz-neprekidna na O i neka je neprekidno diferencijabilna u $x \in O$. Tada je

$$\partial F(x) = \partial_B F(x) = \{\nabla F(x)\}.$$

Dokaz. Uzmimo bilo koji niz $(x_n)_{n \in \mathbb{N}} \subseteq D_F$ takav da $x_n \rightarrow x$. Tada $\nabla F(x_n) \rightarrow \nabla F(x)$ jer je ∇F neprekidna u x . Dakle, $\nabla F(x) \in \partial_B F(x)$.

Neka je sada $G \in \partial_B F$. Tada postoji niz $(x_n)_{n \in \mathbb{N}} \subseteq D_F$ takav da $x_n \rightarrow x$ i $\nabla F(x_n) \rightarrow G$. No, kako je ∇F neprekidna u x , znamo da vrijedi $\nabla F(x_n) \rightarrow \nabla F(x)$. Zbog jedinstvenosti limesa na prostoru matrica slijedi $G = \nabla F(x)$.

Za sada znamo da je $\partial_B F(x) = \{\nabla F(x)\}$. Dodatno,

$$\partial F(x) = \text{co}(\partial_B F(x)) = \text{co}(\{\nabla F(x)\}) = \{\nabla F(x)\}.$$

□

Napomena 3.1.5. Nažalost, u prethodnoj propoziciji ne možemo ispustiti pretpostavku da je ∇F neprekidna u x . Za primjer promotrimo $f: \mathbb{R} \rightarrow \mathbb{R}$ zadanu s

$$f(x) = \begin{cases} x^2 \sin\left(\frac{1}{x}\right), & x \neq 0 \\ 0, & x = 0 \end{cases}.$$

Lako se pokaže da je f lokalno Lipschitz-neprekidna i derivabilna na cijelom \mathbb{R} te da vrijedi

$$f'(x) = \begin{cases} 2x \sin\left(\frac{1}{x}\right) - \cos\left(\frac{1}{x}\right), & x \neq 0 \\ 0, & x = 0 \end{cases}.$$

Promatramo niz $x_n = \frac{1}{2n\pi}$. Vidimo da $x_n \rightarrow 0$ i da je

$$f'(x_n) = \frac{2}{2n\pi} \sin(2n\pi) - \cos(2n\pi) = -1 \rightarrow -1.$$

Za niz $y_n = \frac{1}{(2n-1)\pi}$ imamo $y_n \rightarrow 0$ i $f'(y_n) = 1 \rightarrow 1$ (raspis je analogan). Dakle, vrijedi $x_n \rightarrow 0$ i $y_n \rightarrow 0$, ali $\lim f'(x_n) \neq \lim f'(y_n)$, pa limes $\lim_{x \rightarrow 0} f'(x)$ ne postoji. To također povlači da f' nije neprekidna u 0.

Uočimo da smo dodatno pokazali i ono što je ustvari bitno za ovu napomenu; imamo $f'(0) = 0 \neq 1$, ali $1 \in \partial_B f(0)$. Dakle, $\partial_B f(0) \neq \{0\} = \{f'(0)\}$.

Dodatno uvodimo još jedan pojam koji će nam biti bitan poslije kada uvedemo pojam poluglatke funkcije.

Definicija 3.1.6. Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$, gdje je O otvoren, lokalno Lipschitz-neprekidna na O . Definiramo *derivaciju funkcije F u smjeru vektora d u točki x* , u oznaci $F'(x; d)$, s

$$F'(x; d) := \lim_{t \downarrow 0} \frac{F(x + td) - F(x)}{t}.$$

Postoji jako puno rezultata koji nam pomažu u radu s B-subdiferencijalima, generaliziranim diferencijalima i derivacijama funkcija u smjeru vektora. Na primjer, produktno, kvocijentno, i lančano pravilo se mogu poopćiti. Ovdje se time nećemo baviti, ali ističemo da se ti rezultati mogu pronaći u [4] i [1].

Uveli smo ∂F s ciljem da poopćimo ∇F . To nas motivira da poopćimo Algoritam 2.1.1 na sličan način.

Algoritam 3.1.7 (Newtonova metoda za neglatke sustave). Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je O otvoren, lokalno Lipschitz-neprekidna i $x_0 \in O$. Stavimo $k := 0$.

(1) Ako uvjet zaustavljanja nije zadovoljen, riješi

$$G(x_k)d_k = -F(x_k)$$

po d_k , gdje je $G(x_k)$ po volji izabran element od $\partial F(x_k)$.

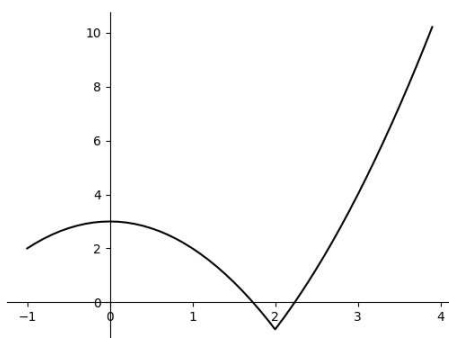
(2) Stavi $x_{k+1} := x_k + d_k$ i $k := k + 1$, te se vrati na (1).

Kao i u Algoritmu 2.1.1, da bismo dokazali da je niz $(x_k)_{k \in \mathbb{N}_0}$ dobro definiran, trebamo pokazati da je $G(x_k)$ invertibilna matrica. No, ovdje je to puno teže. Naime, $\nabla F(x_k)$ je **skup** iz kojeg biramo $G(x_k)$ po volji, te ne možemo tvrditi da $\|G(y) - G(x)\|$ teži u nulu kada $y \rightarrow x$.

Primjer 3.1.8. Promatrajmo funkciju

$$f(x) = |x^2 - 4| - 1.$$

Njene nultočke su $\pm\sqrt{5}$ i $\pm\sqrt{3}$. Dio njenog grafa je na Slici 3.1.



Slika 3.1: Dio grafa funkcije $f(x) = |x^2 - 4| - 1$

Iz grafa se lako vidi da f nije derivabilna u $x = 2$. Dodatno, zbog simetričnosti znamo da nije derivabilna ni u $x = -2$. Pokazuje se da je za ostale točke derivabilna, te da za $x \neq 2$ i $x \neq -2$ vrijedi

$$f'(x) = 2x \operatorname{sgn}(x^2 - 4),$$

gdje je sgn takozvana funkcija predznaka.

Primijenimo sada 10 iteracija Algoritma 3.1.7 s $x_0 = 2$. Ovdje odmah vidimo da se Algoritam 2.1.1 ne može primijeniti jer f nije diferencijabilna u $x_0 = 2$. Slično kao u Primjeru 3.1.3 dobijemo da je $\partial_B f(2) = \{-4, 4\}$ te da je $\partial f(2) = [-4, 4]$.

Najprije probajmo uzeti $G(2) = 4$. U ostalim točkama niza iteracija je funkcija f diferencijabilna pa po Propoziciji 3.1.4 znamo da za G koristimo derivaciju od f u tim točkama. Nakon 10 iteracija za aproksimaciju nultočke imamo 2.23606797749979 što je poprilično dobra aproksimacija za $\sqrt{5}$, za koju znamo da je jedna nultočka od f .

Sada uzmimo $G(2) = -4$. Sličnim postupkom kao u prethodnom odlomku, nakon 10 iteracija dobijemo 1.73205080756888 što je poprilično dobra aproksimacija za $\sqrt{3}$, drugu nultočku od f .

U ovom primjeru vidimo koliko je općenito izbor $G(x_k) \in \partial F(x_k)$ ustvari netrivialno pitanje.

Python kod za $G(2) = 4$:

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sym
from sympy import Float

x = sym.Symbol("x", real=True)

f = sym.Abs(x**2 - 4) - 1

df = sym.diff(f, x)

x_plot = np.arange(-1.0, 4, 0.1)
y_plot = np.array([f.subs(x, x_) for x_ in x_plot])

fig, ax = plt.subplots()
ax.plot(x_plot, y_plot, "k-")

ax.spines['left'].set_position('zero')

ax.spines['right'].set_color('none')
ax.yaxis.tick_left()

ax.spines['bottom'].set_position('zero')

ax.spines['top'].set_color('none')
ax.xaxis.tick_bottom()

plt.show()
```



```

x_approx = [2]

d = -f.subs(x, 2) / 4

x_approx.append(x_approx[0] + d)

for it in range(1, 10):
    f_ = Float(f.subs(x, x_approx[it]))
    df_ = Float(df.subs(x, x_approx[it]))
    d = -f_ / df_
    x_approx.append(x_approx[it] + d)

print(x_approx)

```

Kod nema komentara jer je sličan kodovima koje smo koristili do sada. Dodatno, ako želimo koristiti neki drugi $G(2)$, treba promijeniti $d = -f.subs(x, 2) / 4$ u $d = -f.subs(x, 2) / G(2)$ (npr. stavimo $d = -f.subs(x, 2) / (-4)$).

3.2 Poluglatkoća, teoremi konvergencije poluglatke Newtonove metode

U ovoj sekciji nam je cilj iznijeti teoreme konvergencije Newtonove metode (Algoritma 3.1.7). Pokazuje se da ćemo morati proučavati nešto užu familiju funkcija. No, najprije, možemo dokazati sljedeću važnu lemu.

Lema 3.2.1. *Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je O otvoren, lokalno Lipschitz-neprekidna. Neka je $x \in O$. Ako je $\partial F(x)$ nesingularan, tj. sve $G \in \partial F(x)$ su invertibilne (nesingularne), onda postoje $\beta, r > 0$ takvi da, za sve $y \in K(x, r)$ i sve $G(y) \in \partial F(y)$, $G(y)$ je invertibilna i vrijedi*

$$\|G(y)^{-1}\| \leq \beta.$$

Dokaz. (Skica). Pokažimo najprije da postoji $r > 0$ takav da je za sve $y \in K(x, r)$ i sve $G(y) \in \partial F(y)$ $G(y)$ invertibilna. Pretpostavimo suprotno, tj. da postoji niz $(x_k)_{k \in \mathbb{N}}$ takav da $x_k \rightarrow x$ i postoje $G_k \in \partial F(x_k)$ koje nisu invertibilne. Može se pokazati da, kako $x_k \rightarrow x$, postoji podniz x_{p_k} takav da $G_{p_k} \rightarrow G$. Također se može pokazati da onda mora biti $G \in \partial F(x)$. Dodatno, skup neinvertibilnih matrica je zatvoren skup (jer je det neprekidna funkcija i skup neinvertibilnih matrica je određen s $\det^{-1}(\{0\})$), pa onda G mora biti neinvertibilna (jer $G_{p_k} \rightarrow G$). No, to je kontradikcija s nesingularnošću od $\partial F(x)$.

Pokažimo sada postojanje $\beta > 0$ iz iskaza leme. Pretpostavimo suprotno, tj. da takav β ne postoji. Tada postoji niz $(x_k)_{k \in \mathbb{N}}$ takav da $x_k \rightarrow x$ i postoje $G_k \in \partial F(x_k)$ takve da $\|G_k^{-1}\| \rightarrow \infty$. Uzmimo sada opet podniz x_{p_k} za koji $G_{p_k} \rightarrow G \in \partial F(x)$. Može se pokazati da tada $G_{p_k}^{-1} \rightarrow G^{-1}$, pa $\|G_{p_k}^{-1}\| \rightarrow \|G^{-1}\|$ (zbog neprekidnosti norme), što je kontradikcija s $\|G_k^{-1}\| \rightarrow \infty$. \square

Nažalost, ovo još uvijek nije dovoljno da pokažemo da je Algoritam 3.1.7 dobro definiran. Zato uvodimo sljedeći pojam.

Definicija 3.2.2. Neka je $O \subseteq \mathbb{R}^n$ otvoren i neprazan i $x \in O$. Za funkciju $F: O \rightarrow \mathbb{R}^m$ kažemo da je *poluglatka* u x ako je lokalno Lipschitz-neprekidna u x i ako

$$\lim_{\substack{G \in \partial F(x+t\tilde{d}) \\ \tilde{d} \rightarrow d, t \downarrow 0}} G\tilde{d}$$

postoji za sve $d \in \mathbb{R}^n$. Ako je F poluglatka u svim $x \in O$, tada kažemo da je F *poluglatka* (na cijelom O).

Definicija, iako ispravna, nije previše operabilna. Zato iskazujemo jedan teorem koji nam olakšava rad s poluglatkoćom.

Teorem 3.2.3. Neka je $O \subseteq \mathbb{R}^n$ otvoren i $F: O \rightarrow \mathbb{R}^m$, te $x \in O$. Sljedeće tvrdnje su ekvivalentne:

- (1) F je poluglatka u x .
- (2) F je lokalno Lipschitz-neprekidna u x , $F'(x; \cdot)$ postoji, te, za sve $G(d) \in \partial F(x + d)$,

$$\lim_{d \rightarrow 0} \frac{\|G(d)d - F'(x; d)\|}{\|d\|} = 0.$$

- (3) F je lokalno Lipschitz-neprekidna u x , $F'(x; \cdot)$ postoji, te, za sve $G(d) \in \partial F(x + d)$,

$$\lim_{d \rightarrow 0} \frac{\|F(x + d) - F(x) - G(d)d\|}{\|d\|} = 0.$$

Također, ako je funkcija glatka, tada je i poluglatka, o čemu preciznije govori sljedeća propozicija.

Propozicija 3.2.4. Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^m$, gdje je O otvoren, neprekidno diferencijabilna na nekoj okolini točke $x \in O$. Tada je F i poluglatka u x .

Dokaz. S obzirom da je F neprekidno diferencijabilna na nekoj okolini točke x , znamo i da je lokalno Lipschitz-neprekidna u x .

Dodatno, znamo da ako je F diferencijabilna u $x \in O$, onda za sve $d \in \mathbb{R}^n$ postoji $F'(x; d)$ i vrijedi $F'(x; d) = \nabla F(x)d$.

Neka je U okolina od x na kojoj je F neprekidno diferencijabilna. Uzmimo sada $d \in \mathbb{R}^n$ dovoljno mali da vrijedi $x + d \in U$ (ovo ima smisla jer ćemo gledati limes $d \rightarrow 0$). Sada je, po Propoziciji 3.1.4, $\partial F(x + d) = \{\nabla F(x + d)\}$. Dodatno,

$$\begin{aligned} \frac{\|F(x + d) - F(x) - \nabla F(x + d)d\|}{\|d\|} &= \frac{\|F(x + d) - F(x) - \nabla F(x)d + (\nabla F(x) - \nabla F(x + d))d\|}{\|d\|} \\ &\leq \frac{\|F(x + d) - F(x) - \nabla F(x)d\|}{\|d\|} + \frac{\|(\nabla F(x) - \nabla F(x + d))d\|}{\|d\|} \\ &\leq \frac{\|F(x + d) - F(x) - \nabla F(x)d\|}{\|d\|} + \|\nabla F(x) - \nabla F(x + d)\|. \end{aligned}$$

Iz definicije diferencijala slijedi

$$\lim_{d \rightarrow 0} \frac{\|F(x + d) - F(x) - \nabla F(x)d\|}{\|d\|} = 0,$$

a iz neprekidnosti ∇F u x slijedi

$$\lim_{d \rightarrow 0} \|\nabla F(x) - \nabla F(x + d)\| = 0.$$

Dakle,

$$\lim_{d \rightarrow 0} \frac{\|F(x + d) - F(x) - \nabla F(x + d)d\|}{\|d\|} = 0,$$

pa iz Teorema 3.2.3, tvrdnje (3), slijedi da je F poluglatka u x . \square

Primjer 3.2.5. Promatrajmo opet $f: \mathbb{R} \rightarrow \mathbb{R}$, $f(x) = |x|$. Ona je neprekidno diferencijabilna za sve $x \neq 0$ pa je u njima i poluglatka po Propoziciji 3.2.4. Za $x = 0$ moramo napraviti detaljniju analizu. Najprije uočimo da je f (lokalno) Lipschitz-neprekidna u 0 : za sve $x \in \mathbb{R}$ imamo

$$|f(0) - f(x)| = |f(x)| = |x| = |0 - x|.$$

Uzmimo $d \in \mathbb{R}$. Za $d = 0$ imamo

$$f'(x; 0) = \lim_{t \downarrow 0} \frac{f(0 + t \cdot 0) - f(0)}{t} = 0,$$

za $d > 0$ imamo

$$f'(x; d) = \lim_{t \downarrow 0} \frac{f(td)}{t} = \lim_{t \downarrow 0} \frac{|td|}{t} = \lim_{t \downarrow 0} \frac{td}{t} = d,$$

a za $d < 0$ imamo

$$f'(x; d) = \lim_{t \downarrow 0} \frac{f(td)}{t} = \lim_{t \downarrow 0} = \frac{|td|}{t} = \lim_{t \downarrow 0} \frac{-td}{t} = -d.$$

Dakle, $f'(0; \cdot)$ postoji.

Po Propoziciji 3.1.4 za $d \neq 0$ imamo $\partial f(d) = \{f'(d)\} = \{\text{sgn}(d)\}$. Za $d > 0$ je

$$\frac{|f(d) - f(0) - f'(d)d|}{|d|} = \frac{\|d\| - \text{sgn}(d)d}{|d|} = \frac{|d - d|}{|d|} = 0,$$

a za $d < 0$ je

$$\frac{|f(d) - f(0) - f'(d)d|}{|d|} = \frac{\|d\| - \text{sgn}(d)d}{|d|} = \frac{|-d + d|}{|d|} = 0.$$

Dakle,

$$\lim_{d \rightarrow 0} \frac{|f(d) - f(0) - f'(d)d|}{|d|} = 0,$$

te po Teoremu 3.2.3, tvrdnji (3), zaključujemo da je f poluglatka i u 0.

Konačno, za poluglatke funkcije možemo dokazati lokalnu konvergenciju Algoritma 3.1.7. Dodatno, kada god je F u Algoritmu 3.1.7 poluglatka, taj algoritam nazivamo *poluglatka Newtonova metoda*.

Teorem 3.2.6. *Neka je dana $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je O otvoren. Neka je $x_* \in O$ takav da je $F(x_*) = 0$, F je lokalno Lipschitz-neprekidna i poluglatka u x_* , te neka je $\partial F(x_*)$ nesingularan. Tada postoji $\varepsilon > 0$ takav da je za sve $x_0 \in K(x_*, \varepsilon)$ niz $(x_k)_{k \in \mathbb{N}_0}$ dobiven Algoritmom 3.1.7 (tj. poluglatkom Newtonovom metodom) dobro definiran, konvergira u x_* , i vrijedi*

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = 0. \quad (3.1)$$

Dokaz. Po Lemi 3.2.1 postoje $r, \beta > 0$ takvi da je $\partial F(x)$ nesingularan za sve $x \in K(x_*, r)$, te da za sve $x \in K(x_*, r)$ i sve $G(x) \in \partial F(x)$ vrijedi $\|G(x)^{-1}\| \leq \beta$.

Uzmimo sada $\varepsilon \leq r$ i fiksirajmo $0 < c < 1$. Neka je $G(x_0) \in \partial F(x_0)$. Tada je

$$\begin{aligned} x_1 - x_* &= x_0 - x_* - G(x_0)^{-1}F(x_0) \\ &= x_0 - x_* + G(x_0)^{-1}(F(x_*) - F(x_0)) \\ &= G(x_0)^{-1}(F(x_*) - F(x_0) - G(x_0)(x_* - x_0)). \end{aligned}$$

Kako je F poluglatka u x_* , za $\eta := \frac{c}{\beta} > 0$ postoji $\tilde{r} > 0$ takav da za sve $x \in K(x_*, \tilde{r})$ vrijedi

$$\|F(x_*) - F(x_0) - G(x_0)(x_* - x_0)\| \leq \eta \|x_0 - x_*\|$$

(direktna posljedica Teorema 3.2.3, tvrdnje (3)). Ukoliko je potrebno, smanjimo ε tako da vrijedi $\varepsilon \leq \min\{r, \tilde{r}\}$. Tada imamo

$$\begin{aligned} \|x_1 - x_*\| &= \|G(x_0)^{-1}(F(x_*) - F(x_0) - G(x_0)(x_0 - x_*))\| \\ &\leq \|G(x_0)^{-1}\| \cdot \|F(x_*) - F(x_0) - G(x_0)(x_0 - x_*)\| \\ &\leq \beta \cdot \eta \|x_0 - x_*\| \\ &= c \|x_0 - x_*\| \\ &< \|x_0 - x_*\| \\ &< \varepsilon. \end{aligned}$$

Dakle, $x_1 \in K(x_*, \varepsilon)$. Analogno, indukcijom se pokaže da je $x_k \in K(x_*, \varepsilon)$ i da vrijedi

$$\|x_{k+1} - x_*\| \leq c \|x_k - x_*\|,$$

iz čega slijedi $\lim_{k \rightarrow \infty} x_k = x_*$ (argumenti su isti kao u dokazu teorema lokalne konvergencije glatke Newtonove metode).

Dodatno, iz

$$\begin{aligned} \|x_{k+1} - x_*\| &\leq \|G(x_0)^{-1}\| \cdot \|F(x_*) - F(x_k) - G(x_k)(x_* - x_k)\| \\ &\leq \beta \|F(x_*) - F(x_k) - G(x_k)(x_* - x_k)\| \end{aligned}$$

slijedi i

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \leq \beta \cdot \frac{\|F(x_*) - F(x_k) - G(x_k)(x_* - x_k)\|}{\|x_k - x_*\|} \rightarrow 0$$

kada $k \rightarrow \infty$. □

Primijetimo da (3.1) povlači da je konvergencija poluglatke Newtonove metode linearna. Ustvari, vidimo da bi konvergencija mogla biti i „malo” brža. Ovakve konvergencije nazivamo *superlinearnima*.

Iskažimo (bez dokaza) još i teorem globalne konvergencije.

Teorem 3.2.7. *Neka je $F: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$, gdje je O otvoren, lokalno Lipschitz-neprekidna i poluglatka na U_0 , zatvorenju kugle $K(x_0, r)$. Također, neka je za sve $G(x) \in \partial F(x)$, $x \in U_0$, $G(x)$ invertibilna i, za $y \in U_0$, vrijedi*

$$\begin{aligned} \|G(x)^{-1}\| &\leq \beta, \\ \|G(x)(y - x) - F'(x; y - x)\| &\leq L \|y - x\|, \\ \|F(y) - F(x) - F'(x; y - x)\| &\leq \eta \|y - x\|, \end{aligned}$$

gdje je $\alpha := \beta(L + \eta) < 1$ i $\beta \|F(x_0)\| \leq r(1 - \alpha)$. Tada niz iteracija $(x_k)_{k \in \mathbb{N}_0}$ dobiven primjenom poluglatke Newtonove metode ostaje u U_0 i konvergira prema jedinstvenoj nultočki x_* od F u U_0 . Dodatno,

$$\|x_k - x_*\| \leq \frac{\alpha}{1 - \alpha} \|x_k - x_{k-1}\|, \quad k \in \mathbb{N}.$$

3.3 Poluglatka Newtonova metoda u Banachovim prostorima

Promatramo funkcije $F: D \subseteq X \rightarrow Z$, gdje su X i Z Banachovi prostori. Do sada smo za algoritme Newtonove metode imali $X = Z = \mathbb{R}^n$. Cilj je poopćiti Newtonovu metodu na širu klasu Banachovih prostora.

Najprije nam je cilj poopćiti pojam diferencijabilnosti, da obuhvatimo što širu klasu funkcija na koje možemo primijeniti (poopćenu) Newtonovu metodu. Vidimo da je u dokazu Teorema 3.2.6 svojstvo iz Teorema 3.2.3 (3) bilo ključno. Motivirani time, uvodimo sljedeću definiciju.

Definicija 3.3.1. Neka su X i Z Banachovi prostori. Za preslikavanje $F: D \subseteq X \rightarrow Z$ kažemo da je *generalizirano diferencijabilno* (ili *Newton diferencijabilno*) na otvorenom skupu $O \subseteq D$ ako postoji familija preslikavanja $G: O \rightarrow \mathcal{L}(X, Z)$ takva da je

$$\lim_{d \rightarrow 0} \frac{1}{\|d\|} \|F(x+d) - F(x) - G(x+d)d\| = 0, \quad (3.2)$$

za sve $x \in O$.

Sa $\mathcal{L}(X, Z)$ smo označili skup neprekidnih linearnih operatora sa X u Z . Konačno, možemo opisati poopćenje Newtonove metode.

Algoritam 3.3.2 (Poluglatka Newtonova metoda u Banachovim prostorima). Neka je $F: D \subseteq X \rightarrow Z$ generalizirano diferencijabilno preslikavanje na $O \subseteq D$, $x_0 \in O$, $k := 0$.

(1) Ako kriterij zaustavljanja nije zadovoljen, riješi

$$G(x_k)d_k = -F(x_k)$$

po d_k , gdje je $G(x_k)$ neki (proizvoljni) generalizirani diferencijal od F u x_k .

(2) Stavi $x_{k+1} := x_k + d_k$, $k := k + 1$, i vrati se na (1).

Za prethodno definiran algoritam postoji i teorem lokalne konvergencije.

Teorem 3.3.3. Neka je x_* rješenje od $F(x) = 0$ i neka je F generalizirano diferencijabilno preslikavanje na nekoj otvorenoj okolini O od x_* . Nadalje, neka je $G(x)$ nesingularan za sve $x \in O$ i neka je $\{\|G(x)\| \mid x \in O\}$ omeđen. Tada je Algoritam 3.3.2 dobro definiran i konvergira superlinearno prema x_* ako je x_0 dovoljno blizu x_* , tj. ako je $\|x_0 - x_*\|$ dovoljno malo.

Dokaz je sličan dokazu Teorema 3.2.6.

3.4 Primjeri — numeričko rješavanje ODJ

Pretpostavimo da trebamo riješiti početnu (Cauchyjevu) zadaću za ODJ prvog reda zadanu u obliku

$$y' = f(x, y), \quad y(x_0) = y_0. \quad (3.3)$$

Neka je $[a, b]$ neki segment koji sadrži x_0 . Znamo da je $C[a, b]$ s normom

$$\|f\|_C = \max_{x \in [a, b]} |f(x)|$$

Banachov prostor. Znamo i da je $C^1[a, b]$ s normom

$$\|f\|_{C^1} = \max_{x \in [a, b]} |f(x)| + \max_{x \in [a, b]} |f'(x)| = \|f\|_C + \|f'\|_C$$

Banachov prostor. Dodatno, $(\mathbb{R}, |\cdot|)$ je isto Banachov prostor, pa je i $\mathbb{R} \times C[a, b]$ s normom

$$\|(x, f)\| = |x| + \|f\|_C$$

Banachov prostor.

Sada definiramo

$$F: C^1[a, b] \rightarrow \mathbb{R} \times C[a, b], \quad F(y) = (y(x_0) - y_0, y' - f(\cdot, y))$$

i vidimo da je y rješenje početne zadaće (3.3) ako i samo ako je y nultočka od F . Drugim riječima, ako je F generalizirano diferencijabilno preslikavanje, možemo koristiti Algoritam 3.3.2 da pokušamo aproksimirati rješenje ODJ! („Pokušati aproksimirati” zato što je još otvoreno pitanje konvergencije i dobre definiranosti niza iteracija dobiven uporabom Algoritma 3.3.2.)

No, mi ćemo se ipak poslužiti tehnikom *diskretizacije* da bismo aproksimirali rješenje. Razlog tome je što je rješavanje jednadžbi $G(y_k)d_k = -F(y_k)$ (iz koraka (1) Algoritma 3.3.2) i dalje dosta kompleksan problem. Ipak, ovo ostaje jedan zoran primjer gdje bi se Algoritam 3.3.2 teoretski mogao iskoristiti.

Primjer 3.4.1. Promatrajmo početnu zadaću

$$y' = y, \quad y(0) = 1. \quad (3.4)$$

Naravno, znamo njeno rješenje: $y(x) = e^x$. Ovdje ćemo pokušati aproksimirati njeno rješenje na segmentu $[0, 1]$. Definirajmo

$$F: C^1[0, 1] \rightarrow \mathbb{R} \times C[0, 1], \quad F(y) = (y(0) - 1, y' - y)$$

i pokažimo da je F generalizirano diferencijabilna na cijeloj svojoj domeni.

Uzmimo $y, d \in C^1[0, 1]$. Tada je

$$\begin{aligned} F(y + d) - F(y) &= ((y + d)(0) - 1, (y + d)' - (y + d)) - (y(0) - 1, y' - y) \\ &= (y(0) + d(0) - 1, y' + d' - y - d) - (y(0) - 1, y' - y) \\ &= (d(0), d' - d). \end{aligned}$$

Neka je $z \in C^1[0, 1]$. Definirajmo

$$G(z)d := (d(0), d' - d).$$

Primijetimo da je to uistinu linearni operator sa $C^1[0, 1]$ u $\mathbb{R} \times C[0, 1]$:

$$\begin{aligned} G(z)(\alpha d_1 + \beta d_2) &= ((\alpha d_1 + \beta d_2)(0), (\alpha d_1 + \beta d_2)' - (\alpha d_1 + \beta d_2)) \\ &= (\alpha d_1(0) + \beta d_2(0), \alpha d_1' + \beta d_2' - \alpha d_1 - \beta d_2) \\ &= \alpha(d_1(0), d_1' - d_1) + \beta(d_2(0), d_2' - d_2) \\ &= \alpha G(z)d_1 + \beta G(z)d_2. \end{aligned}$$

Dodatno,

$$\begin{aligned} \|G(z)d\| &= \|(d(0), d - d')\| = |d(0)| + \|d - d'\|_C \\ &\leq \|d\|_C + \|d\|_C + \|d'\|_C \\ &\leq 2(\|d\|_C + \|d'\|_C) \\ &= 2\|d\|_{C^1} \end{aligned}$$

(iskoristili smo $|d(0)| \leq \|d\|_C$ i nejednakost trokuta norme u prvoj nejednakosti, a u drugoj činjenicu da je $\|d'\|_C \leq 2\|d'\|_C$), iz čega zaključujemo da je $G(z)$ i ograničen linearni operator, tj. $G(z) \in \mathcal{L}(C^1[0, 1], \mathbb{R} \times C[0, 1])$.

Konačno, sada imamo

$$\lim_{d \rightarrow 0} \frac{\|F(y + d) - F(y) - G(y + d)d\|}{\|d\|_{C^1}} = \lim_{d \rightarrow 0} \frac{\|(d(0), d' - d) - (d(0), d' - d)\|}{\|d\|_{C^1}} = 0,$$

što potvrđuje da je F generalizirano diferencijabilna.

Kako sada upotrijebiti Newtonovu metodu? Morat ćemo funkciju *diskretizirati* da bismo mogli računati s računalom i numeričkim paketima. Drugim riječima, segment $[0, 1]$ ćemo podijeliti na n jednakih dijelova širine $h := \frac{1}{n}$. Pišemo y_i umjesto $y(ih)$. Dodatno, pitanje je što ćemo sa y' . Derivaciju ćemo aproksimirati u točkama ih , $i = 1, 2, \dots, n$, s

$$y'(ih) \approx \frac{y(ih) - y((i-1)h)}{h} = \frac{y_i - y_{i-1}}{h}.$$

Uočimo da **ne** aproksimiramo $y'(0)$! Nju nećemo ni razmatrati u jednadžbama.

Umjesto F koju smo razmatrali do sada, razmatrat ćemo njenu diskretiziranu varijantu (koju ćemo isto označavati s F):

$$F(y_0, y_1, \dots, y_n) = \begin{bmatrix} y_0 - 1 \\ \frac{y_1 - y_0}{h} - y_1 \\ \frac{y_2 - y_1}{h} - y_2 \\ \vdots \\ \frac{y_n - y_{n-1}}{h} - y_n \end{bmatrix}.$$

To je funkcija s \mathbb{R}^{n+1} u \mathbb{R}^{n+1} . I upravo ćemo na nju primijeniti Newtonovu metodu, tj. tražit ćemo njenu nultočku!

Naravno, ovo je puno jednostavnija situacija, primijenjujemo dobro poznatu realnu glatku Newtonovu metodu, i odmah znamo da je diferencijal od F određen Jacobijevom matricom:

$$\nabla F(y_0, y_1, \dots, y_n) = \left[\frac{\partial F_i}{\partial y_j} \right]_{ij} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -\frac{1}{h} & \frac{1}{h} - 1 & 0 & \dots & 0 \\ 0 & -\frac{1}{h} & \frac{1}{h} - 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{h} - 1 \end{bmatrix}.$$

Nećemo koristiti implementaciju koju smo već napravili. Razlog tomu je što se ovdje koristi jako puno varijabli i simboličko računanje je jednostavno presporo. Dodajmo još da za uvjet zaustavljanja koristimo $\|y^{(n+1)} - y^{(n)}\| < \varepsilon$, gdje je $y^{(n)}$ n -ta iteracija Newtonove metode, za neki određeni ε (u kodu `tol`), ali i da je maksimalan broj iteracija prije „nasilnog” zaustavljanja 100. Početna aproksimacija nam je $y^{(0)} = (1, 1, \dots, 1)$.

Slika 3.2 prikazuje aproksimaciju rješenja, i vidimo da se dosta dobro poklapa s e^x . Istaknimo da su bile potrebne samo dvije iteracije i da je $\|F(y^{(2)})\| \approx 3.78 \times 10^{-12}$. Python kod:

```
import numpy as np
import matplotlib.pyplot as plt

# Koliko segmenata?
n = 1000
h = 1/n

# Ovdje spremamo aproksimaciju rjesenja.
y = np.ones(n+1)
```

```
# Epsilon u tekstu primjera.
tol = 1e-12

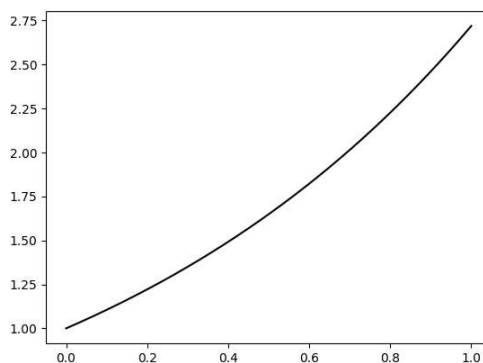
for it in range(0, 100):
    F = np.zeros(n+1)
    F[0] = y[0] - 1
    for i in range(1, n+1):
        F[i] = (y[i] - y[i-1])/h - y[i]

    # J je Jacobijeva matrica od F.
    J = np.zeros((n+1, n+1))
    J[0,0] = 1
    for i in range(1, n+1):
        J[i, i-1] = -1/h
        J[i, i] = 1/h - 1

    # Numericki je bolje rjesavati  $Jd = F$  nego racunati
    #  $J^{-1}$  i onda  $d = J^{-1}F$ .
    d = np.linalg.solve(J, F)
    y = y - d
    if np.linalg.norm(d) < tol:
        break

# Mreza da nacrtamo rjesenje.
t = np.linspace(0, 1, num=n+1)

# Crtamo aproksimaciju rjesenja.
plt.plot(t, y, "k")
plt.show()
```



Slika 3.2: Numeričko rješenje početne zadaće (3.4)

Primjer 3.4.2. Za malo kompliciraniji primjer pogledajmo

$$y' = \sqrt{\ln x} + \cos y, \quad y(3) = 12 \quad (3.5)$$

i aproksimirajmo njeno rješenje na $[3, 8]$. Također, neka je $g: [3, 8] \rightarrow \mathbb{R}$, $g(x) = \sqrt{\ln x}$.

Definiramo

$$F: C^1[3, 8] \rightarrow \mathbb{R} \times C[3, 8], \quad F(y) = (y(3) - 12, y' - g - \cos y).$$

Tada je

$$\begin{aligned} F(y + d) - F(y) &= (d(3), d' - \cos(y + d) + \cos y) \\ &= (d(3), d' - \cos y \cos d + \sin y \sin d + \cos y) \\ &= (d(3), d' + \cos y(1 - \cos d) + \sin y \sin d). \end{aligned}$$

Za $z \in C^1[3, 8]$ definirajmo

$$G(z)d := (d(3), d' + d \sin z),$$

očito je $G(z)$ linearni operator s $C^1[3, 8]$ u $\mathbb{R} \times C[3, 8]$, te vrijedi

$$\begin{aligned} \|G(z)d\| &= |d(3)| + \|d' + d \sin z\|_C \\ &\leq \|d\|_C + \|d'\|_C + \|d \sin z\|_C \\ &\leq \|d\|_C + \|d'\|_C + \|d\|_C \|\sin z\|_C \\ &\leq 2\|d\|_C + \|d'\|_C \\ &\leq 2(\|d\|_C + \|d'\|_C) \\ &= 2\|d\|_{C^1}, \end{aligned}$$

gdje smo, uz tehnike korištene u Primjeru 3.4.1, dodatno koristili $\|\sin z\|_C \leq 1$ i $\|fg\|_C \leq \|f\|_C \|g\|_C$. Dakle, $G(z)$ je i ograničeni linearni operator.

Dokažimo da je F generalizirano diferencijabilna. Dobivamo da je

$$\begin{aligned} & \frac{\|F(y+d) - F(d) - G(y+d)d\|}{\|d\|_{C^1}} \\ &= \frac{\|(d(3), d' + \cos y(1 - \cos d) + \sin y \sin d) - (d(3), d' + d \sin(y+d))\|}{\|d\|_{C^1}} \\ &= \frac{\|(0, \cos y(1 - \cos d) + \sin y \sin d - d \sin y \cos d - d \cos y \sin d)\|}{\|d\|_{C^1}} \\ &= \frac{\|\cos y(1 - \cos d - d \sin d) + \sin y(\sin d - d \cos d)\|_C}{\|d\|_{C^1}} \\ &\leq \frac{\|\cos y\|_C \|1 - \cos d - d \sin d\|_C + \|\sin y\|_C \|\sin d - d \cos d\|_C}{\|d\|_{C^1}} \end{aligned}$$

Uočimo i da je $\|d\|_{C^1} \geq \|d\|_C$ i da je $\|f\|_C = \|\frac{f}{d} \cdot d\|_C \leq \|\frac{f}{d}\|_C \|d\|_C$, pa je

$$\begin{aligned} & \frac{\|F(y+d) - F(d) - G(y+d)d\|}{\|d\|_{C^1}} \\ &\leq \|\cos y\|_C \left\| \frac{1 - \cos d - d \sin d}{d} \right\|_C + \|\sin y\|_C \left\| \frac{\sin d - d \cos d}{d} \right\|_C \\ &= \|\cos y\|_C \left\| \frac{1 - \cos d}{d} - \sin d \right\|_C + \|\sin y\|_C \left\| \frac{\sin d}{d} - \cos d \right\|_C. \end{aligned}$$

Neka je $\varepsilon > 0$ proizvoljan. Znamo da je, gledajući realne limese,

$$\lim_{x \rightarrow 0} \left[\frac{1 - \cos x}{x} - \sin x \right] = 0, \quad \lim_{x \rightarrow 0} \left[\frac{\sin x}{x} - \cos x \right] = 0,$$

pa postoji $\delta > 0$ takav da je za sve $|x| < \delta$

$$\left| \frac{1 - \cos x}{x} - \sin x \right| < \varepsilon, \quad \left| \frac{\sin x}{x} - \cos x \right| < \varepsilon.$$

Ako sad uzmemo d takav da je $|d(x)| < \delta$ za sve $x \in [3, 8]$ (a to ima smisla jer gledamo limes kada $d \rightarrow 0$), imamo

$$\left| \frac{1 - \cos d(x)}{d(x)} - \sin d(x) \right| < \varepsilon, \quad x \in [3, 8] \implies \left\| \frac{1 - \cos d}{d} - \sin d \right\|_C < \varepsilon,$$

a slično dobivamo i

$$\left\| \frac{\sin d}{d} - \cos d \right\|_C < \varepsilon.$$

Dakle,

$$\frac{\|F(y+d) - F(d) - G(y+d)d\|}{\|d\|_{C^1}} < \varepsilon(\|\cos y\|_C + \|\sin y\|_C).$$

Drugim riječima, možemo postići da je $\frac{\|F(y+d) - F(d) - G(y+d)d\|}{\|d\|_{C^1}}$ proizvoljno malo tako da „smanjujemo” d , pa je

$$\lim_{d \rightarrow 0} \frac{\|F(y+d) - F(d) - G(y+d)d\|}{\|d\|_{C^1}} = 0.$$

Konačno, primijenimo Newtonovu metodu. Opet diskretiziramo, tehnika je ista kao u Primjeru 3.4.1. Imamo $y_i = y(3 + ih)$. Diskretizacija od F je sada dana s

$$F(y_0, y_1, \dots, y_n) = \begin{bmatrix} \frac{y_1 - y_0}{h} - \sqrt{\ln(3+h)} - \cos y_1 \\ \frac{y_2 - y_1}{h} - \sqrt{\ln(3+2 \cdot h)} - \cos y_2 \\ \vdots \\ \frac{y_n - y_{n-1}}{h} - \sqrt{\ln(8)} - \cos y_n \end{bmatrix},$$

Jacobijeva matrica je

$$\nabla F(y_0, y_1, \dots, y_n) = \left[\frac{\partial F_i}{\partial y_j} \right]_{ij} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -\frac{1}{h} & \frac{1}{h} + \sin y_1 & 0 & \dots & 0 \\ 0 & -\frac{1}{h} & \frac{1}{h} + \sin y_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{h} + \sin y_n \end{bmatrix}.$$

Za početnu aproksimaciju smo opet uzeli $y^{(0)} = (1, 1, \dots, 1)$. Bilo je potrebno 13 iteracija, a $\|F(y^{(13)})\| \approx 5.5 \times 10^{-11}$. Aproksimacija rješenja se može vidjeti na Slici 3.3. Python kod:

```
import numpy as np
import matplotlib.pyplot as plt

# Koliko segmenata?
n = 5000
h = 5/n

# Ovdje spremamo aproksimaciju rjesenja.
y = np.ones(n+1)

# Epsilon u tekstu primjera.
```

```
tol = 1e-12

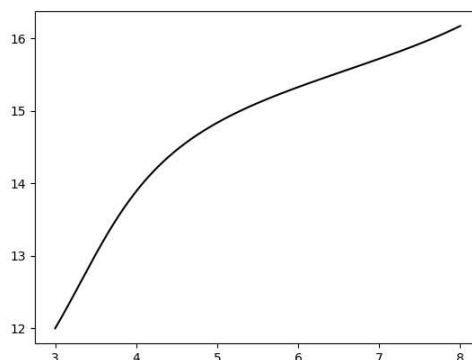
for it in range(0, 100):
    F = np.zeros(n+1)
    F[0] = y[0] - 12
    for i in range(1, n+1):
        temp = np.sqrt(np.log(3+i*h)) + np.cos(y[i])
        F[i]=(y[i]-y[i-1])/h - temp

    # J je Jacobijeva matrica od F.
    J = np.zeros((n+1, n+1))
    J[0,0] = 1
    for i in range(1, n+1):
        J[i, i-1] = -1/h
        J[i, i] = 1/h + np.sin(y[i])

    # Numericki je bolje rjesavati  $Jd = F$  nego racunati
    #  $J^{-1}$  i onda  $d = J^{-1}F$ .
    d = np.linalg.solve(J, F)
    y = y - d
    if np.linalg.norm(d) < tol:
        break

# Mreza da nacrtamo rjesenje.
t = np.linspace(3, 8, num=n+1)

# Crtamo aproksimaciju rjesenja.
plt.plot(t, y, "k")
plt.show()
```



Slika 3.3: Numeričko rješenje početne zadaće (3.5)

Naravno, rješenja zadaća za ODJ drugog reda možemo aproksimirati na sličan način. Jedino se treba sjetiti da je $C^2[a, b]$ s normom

$$\|f\|_{C^2} = \|f\|_C + \|f'\|_C + \|f''\|_C$$

Banachov prostor. U sljedećim primjerima se bavimo ODJ drugog reda s *Dirichletovim rubnim uvjetima*.

Primjer 3.4.3. Promatrajmo zadaću

$$y'' + \sin y = 0, \quad y(0) = 0, \quad y(10) = 0.1. \quad (3.6)$$

Analitički možemo riješiti tek njenu aproksimaciju:

$$y'' + y = 0, \quad y(0) = 0, \quad y(10) = 0.1. \quad (3.7)$$

(aproksimirali smo $\sin y \approx y$). Lako dobijemo analitičko rješenje od (3.7):

$$y(x) = \frac{0.1}{\sin 10} \sin(x).$$

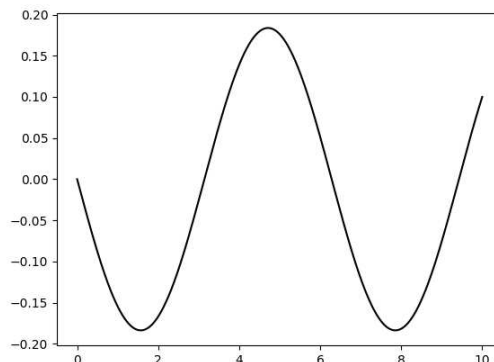
Definiramo $F: C^2[0, 10] \rightarrow \mathbb{R} \times \mathbb{R} \times C[0, 10]$ s

$$F(y) = (y(0), y(10) - 0.1, y'' + y).$$

Analogno kao i prije se pokazuje da je F generalizirano diferencijabilna.

Kod diskretizacije aproksimiramo

$$y''(ih) \approx \frac{y_{i+1} + y_{i-1} - 2y_i}{h^2}, \quad i = 1, 2, \dots, n-1.$$



Slika 3.4: Numeričko rješenje rubne zadaće (3.7)

Diskretizaciju od F zadajemo s $F: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$,

$$F(y_0, y_1, y_2, \dots, y_n) = \begin{bmatrix} y_0 \\ \frac{y_2 + y_0 - 2y_1}{h^2} + y_1 \\ \vdots \\ \frac{y_n + y_{n-2} - 2y_{n-1}}{h^2} + y_{n-1} \\ y_n - 0.1 \end{bmatrix},$$

te je

$$\nabla F(y_0, y_1, \dots, y_n) = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \frac{1}{h^2} & 1 - \frac{2}{h^2} & \frac{1}{h^2} & \cdot & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{h^2} \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Za početnu aproksimaciju smo uzeli $y = (0, 0, \dots, 0)$. Bile su potrebne 4 iteracije i imamo $\|F(y^{(4)})\| \approx 8.1 \times 10^{-12}$. Aproksimacija rješenja se vidi na Slici 3.4, što se dobro poklapa s analitičkim rješenjem. Python kod:

```
import numpy as np
import matplotlib.pyplot as plt
```

```
n = 1000
```

```
h = 10/n
```

```
y = np.zeros(n+1)
```



```

tol = 1e-12

for it in range(0, 100):
    F = np.zeros(n+1)
    F[0] = y[0]
    F[n] = y[n] - 0.1
    for i in range(1, n):
        F[i] = (y[i+1] + y[i-1] - 2 * y[i])/(h**2) + y[i]

    J = np.zeros((n+1, n+1))
    J[0,0] = 1
    for i in range(1, n):
        J[i, i-1] = 1/(h**2)
        J[i, i] = -2/(h**2) + 1
        J[i, i+1] = 1/(h**2)
    J[n,n] = 1

    d = np.linalg.solve(J, F)
    y = y - d
    if np.linalg.norm(d) < tol:
        break

t = np.linspace(0, 10, num=n+1)

plt.plot(t, y, "k")
plt.show()

```

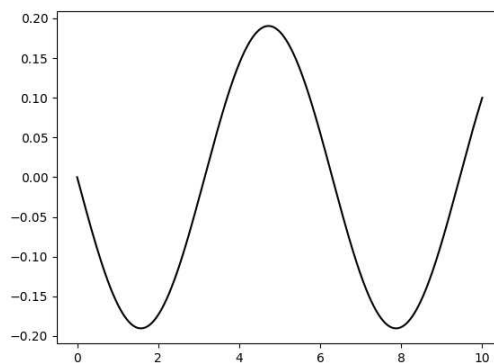
No, lako možemo aproksimirati i rješenje od (3.6). Definiramo $G: C^2[0, 10] \rightarrow \mathbb{R} \times \mathbb{R} \times C[0, 10]$ s

$$G(y) = (y(0), y(10) - 0.1, y'' + \sin y).$$

Analogno kao i prije se pokazuje da je G generalizirano diferencijabilna.

Diskretizaciju od G zadajemo s $G: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$,

$$G(y_0, y_1, y_2, \dots, y_n) = \begin{bmatrix} y_0 \\ \frac{y_2 + y_0 - 2y_1}{h^2} + \sin y_1 \\ \vdots \\ \frac{y_n + y_{n-2} - 2y_{n-1}}{h^2} + \sin y_{n-1} \\ y_n - 0.1 \end{bmatrix},$$



Slika 3.5: Numeričko rješenje rubne zadaće (3.6)

te je

$$\nabla G(y_0, y_1, \dots, y_n) = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ \frac{1}{h^2} & \cos y_1 - \frac{2}{h^2} & \frac{1}{h^2} & \cdot & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{h^2} \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Za početnu aproksimaciju smo uzeli $y = (0, 0, \dots, 0)$. Bilo je potrebno 5 iteracija i imamo $\|G(y^{(5)})\| \approx 8.2 \times 10^{-12}$. Aproksimacija rješenja se vidi na Slici 3.5. Python kod (pišemo F umjesto G):

```
import numpy as np
import matplotlib.pyplot as plt
```

```
n = 1000
h = 10/n
```

```
y = np.zeros(n+1)
```

```
tol = 1e-12
```

```
for it in range(0, 100):
    F = np.zeros(n+1)
    F[0] = y[0]
    F[n] = y[n] - 0.1
    for i in range(1, n):
```

```

s = np.sin(y[i])
F[i] = (y[i+1] + y[i-1] - 2 * y[i])/(h**2) + s

J = np.zeros((n+1, n+1))
J[0,0] = 1
for i in range(1, n):
    J[i, i-1] = 1/(h**2)
    J[i, i] = -2/(h**2) + np.cos(y[i])
    J[i, i+1] = 1/(h**2)
J[n,n] = 1

d = np.linalg.solve(J, F)
y = y - d
if np.linalg.norm(d) < tol:
    break

t = np.linspace(0, 10, num=n+1)

plt.plot(t, y, "k")
plt.show()

```

Označimo sada sa y_1 našu aproksimaciju rješenja zadatke (3.6), a sa y_2 našu aproksimaciju rješenja zadatke (3.7). Tada možemo nacrtati graf od razlike $y_1 - y_2$, što smo napravili na Slici 3.6. Dodatno ističemo da smo (numerički) dobili

$$\|y_1 - y_2\|_C = \max_{x \in [0,10]} |y_1(x) - y_2(x)| \approx 0.007523803393011397,$$

što i nije velika razlika.

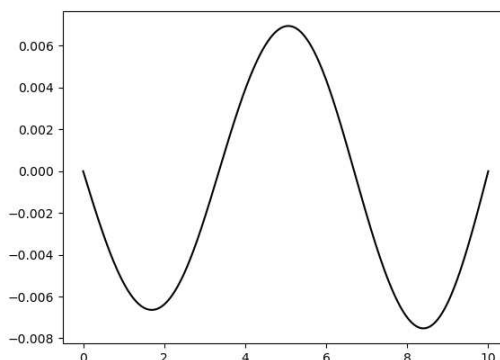
Primjer 3.4.4. Promotrimo zadatak

$$y'' + \sin y = 0, \quad y(0) = 0, \quad y(10) = 0.5 \quad (3.8)$$

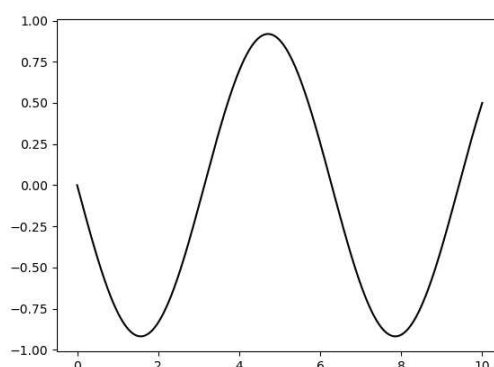
i

$$y'' + y = 0, \quad y(0) = 0, \quad y(10) = 0.5. \quad (3.9)$$

Te zadatke su slične zadatacama iz prethodnog primjera, samo smo promijenili drugi rubni uvjet — umjesto $y(10) = 0.1$ stavili smo $y(10) = 0.5$. Aproksimiramo rješenja tih zadataka na identičan način kao u prošlom primjeru. Aproksimacija rješenja zadatke (3.9) se vidi na Slici 3.7, a aproksimacija rješenja zadatke (3.8) se vidi na Slici 3.8. Ovime je već jasno da je razlika u ovom slučaju ogromna. Python kod je analogan kao i u prošlom primjeru pa ćemo samo opisati razlike. Umjesto $F[n]=y[n]-0.1$ treba staviti $F[n]=y[n]-0.5$.



Slika 3.6: Razlika numeričkih rješenja zadatke (3.6) i linearizirane zadatke (3.7)



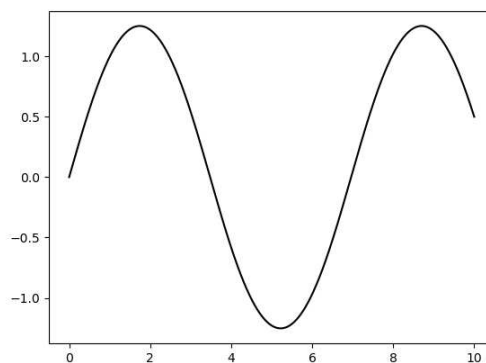
Slika 3.7: Numeričko rješenje rubne zadatke (3.9)

Nadalje, povećali smo maksimalni broj iteracija sa 100 na 10000 (za (3.9) je trebalo 230 iteracija, a za (3.8) čak 2636).

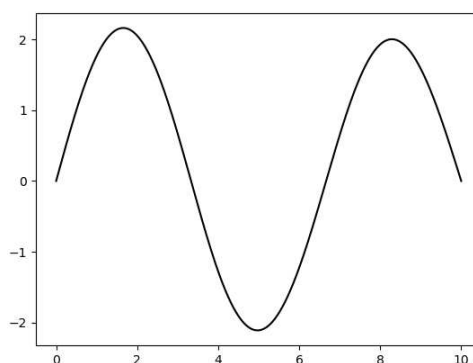
Kao i u prošlom primjeru, označimo sa y_1 našu aproksimaciju rješenja (3.8), a sa y_2 našu aproksimaciju rješenja (3.9). Tada možemo nacrtati graf od razlike $y_1 - y_2$, što smo napravili na Slici 3.9. Dodatno ističemo da smo (numerički) dobili

$$\|y_1 - y_2\|_C = \max_{x \in [0, 10]} |y_1(x) - y_2(x)| \approx 2.1643126028767052,$$

što je stvarno velika razlika.



Slika 3.8: Numeričko rješenje rubne zadaće (3.8)



Slika 3.9: Razlika numeričkih rješenja zadaće (3.8) i linearizirane zadaće (3.9)

Primjer 3.4.5. Promotrimo zadaću

$$y'' = |y|, \quad y(0) = y(1) = 1. \quad (3.10)$$

U ovom primjeru ćemo odmah promatrati diskretizirani problem. Dakle, promatramo funk-

ciju $F: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$ zadanu s

$$F(y_0, y_1, \dots, y_n) = \begin{bmatrix} y_0 - 1 \\ \frac{y_0 + y_2 - 2y_1}{h^2} - |y_1| \\ \frac{y_1 + y_3 - 2y_2}{h^2} - |y_2| \\ \vdots \\ \frac{y_{n-2} + y_n - 2y_{n-1}}{h^2} - |y_{n-1}| \\ y_n - 1 \end{bmatrix}.$$

Uočimo da ovdje imamo mali problem s time što F nije glatka. Zato ćemo morati promatrati generalizirani diferencijal od F .

Neka je $G = [g_{ij}] \in \partial F(y_0, y_1, \dots, y_n)$. Tada imamo

$$\begin{aligned} g_{11} &= g_{n+1, n+1} = 1, \\ g_{1k} &= 0, \quad k = 2, 3, \dots, n+1, \\ g_{k, n+1} &= 0, \quad k = 1, 2, \dots, n, \\ g_{i, i-1} &= g_{i, i+1} = \frac{1}{h^2}, \quad i = 2, 3, \dots, n, \end{aligned}$$

a za g_{ii} , $i = 2, 3, \dots, n$ su stvari malo kompleksnije. Ako je $y_{i-1} > 0$, onda je $g_{ii} = -\frac{2}{h^2} - 1$, ako je $y_{i-1} < 0$, onda je $g_{ii} = -\frac{2}{h^2} + 1$, a ako je $y_{i-1} = 0$, onda je

$$g_{ii} \in \left[-\frac{2}{h^2} - 1, -\frac{2}{h^2} + 1 \right].$$

Zato ćemo za izbor elementa generaliziranog diferencijala, kada za to bude potrebe, koristiti Python funkciju `random.uniform(-1, 1)`, kao što je jasno iz sljedećeg koda:

```
import numpy as np
import matplotlib.pyplot as plt
import random

n = 1000
h = 1/n

y = np.zeros(n+1)

tol = 1e-12
```

```

for it in range(0, 1000):
    F = np.zeros(n+1)
    F[0] = y[0] - 1
    F[n] = y[n] - 1
    for i in range(1, n):
        s = np.abs(y[i])
        F[i] = (y[i+1] + y[i-1] - 2 * y[i])/(h**2) - s

    dF = np.zeros((n+1, n+1))
    dF[0,0] = 1
    for i in range(1, n):
        dF[i, i-1] = 1/(h**2)
        dF[i, i+1] = 1/(h**2)
        if y[i] > 0:
            dF[i, i] = -2/(h**2) - 1
        elif y[i] < 0:
            dF[i, i] = -2/(h**2) + 1
        else :
            dF[i, i] = -2/(h**2) - random.uniform(-1, 1)
    dF[n,n] = 1

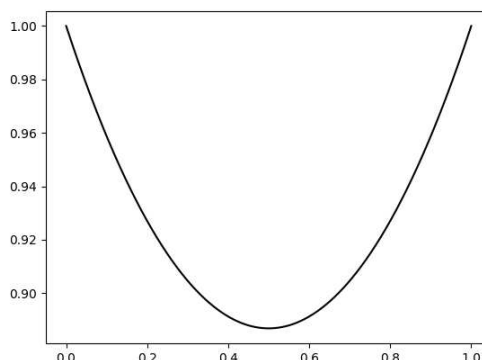
    d = np.linalg.solve(dF, F)
    y = y - d
    if np.linalg.norm(d) < tol:
        break

t = np.linspace(0, 1, num=n+1)

plt.plot(t, y, "k")
plt.show()

```

Nakon nekoliko pokretanja primjećujemo da se broj izvršenih iteracija može značajno razlikovati (npr. pri jednom pokretanju imali smo samo 6 iteracija, dok smo pri jednom drugom pokretanju imali čak 879 iteracija). No, pri svakom pokretanju smo dobili otprilike istu aproksimaciju rješenja koje se može vidjeti na Slici 3.10.



Slika 3.10: Numeričko rješenje rubne zadaće (3.10)

3.5 Konveksni slučaj

U ovoj sekciji ćemo se nakratko pozabaviti jednom važnom klasom realnih funkcija, takozvanim *konveksnim* funkcijama. Cilj nam je dati samo kratak pregled te važne klase funkcija, a za više detalja čitatelj može pogledati [5].

Definicija 3.5.1. Neka je X realni vektorski prostor. Funkciju $f: S \subseteq X \rightarrow \mathbb{R}$, gdje je S konveksan, nazivamo *konveksnom* ako za sve $x, y \in S$ vrijedi

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

za sve $\lambda \in [0, 1]$.

U daljnjem razmatranju ćemo se ograničiti na $X = \mathbb{R}^n$ (više od toga nam za ovu sekciju uistinu nije potrebno). Vrijedi sljedeći važni rezultat.

Teorem 3.5.2. Neka je $O \subseteq \mathbb{R}^n$ otvoren i konveksan i $f: O \rightarrow \mathbb{R}$ diferencijabilna funkcija. Vrijedi da je f konveksna ako i samo ako je

$$f(y) \geq f(x) + \nabla f(x)(y - x)$$

za sve $x, y \in O$.

Intuitivno, prethodni teorem nam govori da je diferencijabilna funkcija f konveksna ako i samo ako joj je graf uvijek iznad tangencijalnih ravnina. Također, motivirani tom važnom karakterizacijom konveksnih funkcija, uvodimo poopćenje diferencijala za konveksne funkcije.

Definicija 3.5.3. Neka je $O \subseteq \mathbb{R}^n$ otvoren i konveksan i neka je $f: O \rightarrow \mathbb{R}^n$ konveksna funkcija. Definiramo *konveksni subdiferencijal* u točki $x \in O$, u oznaci $\partial_C f(x)$, kao

$$\partial_C f(x) := \{l \in \mathbb{R}^n \mid f(y) \geq f(x) + l^T(y - x), \forall y \in O\}.$$

Napomena 3.5.4. U [5] se konveksni subdiferencijal definira kao skup neprekidnih funkcionala. No, iz Rieszovog teorema o reprezentaciji slijedi da za sve $g \in \mathcal{L}(\mathbb{R}^n, \mathbb{R})$ postoji jedinstveni $l \in \mathbb{R}^n$ takav da je

$$g(x) = l^T x, \quad \forall x \in \mathbb{R}^n,$$

pa smo se mi poslužili tom identifikacijom.

Konačno, možemo iznijeti konveksnu verziju Newtonove metode.

Algoritam 3.5.5 (Newtonova metoda za konveksne funkcije). Neka je $O \subseteq \mathbb{R}$ otvoreni interval, $f: O \rightarrow \mathbb{R}$ konveksna funkcija, te $x_0 \in O$. Stavimo $k := 0$.

(1) Ako uvjet zaustavljanja nije zadovoljen, riješi

$$l_k d_k = -f(x_k)$$

po d_k , gdje je l_k po volji izabran element od $\partial_C f(x_k)$.

(2) Stavi $x_{k+1} := x_k + d_k$ i $k := k + 1$, te se vrati na (1).

Koja je veza ove naše digresije s do sad iznesenom teorijom? Za odgovoriti na to pitanje, najprije uvodimo novi pojam.

Definicija 3.5.6. Neka je $f: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, gdje je O otvoren, lokalno Lipschitz-neprekidna. Definiramo *Clarkeovu generaliziranu derivaciju od f u točki $x \in O$ u smjeru vektora $d \in \mathbb{R}^n$* kao

$$f^\circ(x; d) = \limsup_{y \rightarrow x, t \downarrow 0} \frac{f(y + td) - f(y)}{t}.$$

Iskazat ćemo jedan zanimljiv rezultat koji povezuje prethodno uvedeni pojam i generalizirani diferencijal (preciznije, generalizirani gradijent).

Propozicija 3.5.7. Neka je $f: O \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, gdje je O otvoren, lokalno Lipschitz-neprekidna. Tada je

$$\partial f(x) = \{l \in \mathbb{R}^n \mid l^T d \leq f^\circ(x; d), \forall d \in \mathbb{R}^n\}.$$

Dodatno, za sve $(x, d) \in O \times \mathbb{R}^n$ imamo

$$f^\circ(x; d) = \max\{l^T d \mid l \in \partial f(x)\}.$$

Slijedi jedan važan rezultat koji povezuje klase konveksnih i lokalno Lipschitz-neprekidnih funkcija.

Teorem 3.5.8. *Neka je $O \subseteq \mathbb{R}^n$ otvoren i konveksan, te $f: O \rightarrow \mathbb{R}$ konveksna funkcija. Tada je f i lokalno Lipschitz-neprekidna.*

Konačno, sljedeći rezultat predstavlja svojevrsnu spojnicu između ideje iznesene u ovoj sekciji (konveksne Newtonove metode) i ideje s početka poglavlja (neglatke Newtonove metode).

Teorem 3.5.9. *Neka je $O \subseteq \mathbb{R}^n$ otvoren i konveksan, te $f: O \rightarrow \mathbb{R}$ konveksna funkcija. Tada je $f^\circ(x; d) = f'(x; d)$ za sve $(x, d) \in O \times \mathbb{R}^n$ i $\partial f(x) = \partial_C f(x)$ za sve $x \in O$.*

Bibliografija

- [1] F. H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, 1990.
- [2] Z. Drmač, V. Hari, M. Marušić, M. Rogina, S. Singer i S. Singer, *Numerička analiza: Predavanja i vježbe*, 2003.
- [3] O. P. Ferreira i B. F. Svaiter, *Kantorovich's Theorem on Newton's Method*, 2007. (zadnje pristupljeno 1. 9. 2020.), <https://arxiv.org/pdf/1209.5704.pdf>.
- [4] M. Hintermüller, *Semismooth Newton Methods and Applications*, 2010. (zadnje pristupljeno 1. 9. 2020.), https://www.math.uni-hamburg.de/home/hinze/Psfiles/Hintermueller_OWNotes.pdf.
- [5] J. Jahn, *Introduction to the Theory of Nonlinear Optimization*, Springer, 2007.

Sažetak

Ukratko, u ovome radu se bavimo Newtonovom metodom i njenim raznovrsnim poopćenjima.

U prvom poglavlju proučavamo standardnu Newtonovu metodu za glatku realnu funkciju realne varijable. Motiviramo metodu, opisujemo je, iznosimo teoreme konvergencije, izvodimo ocjene greške. Ukratko opisujemo kako se situacija mijenja kada imamo višestruku nultočku. Za kraj, implementiramo ovu metodu u Pythonu i navodimo još nekoliko primjera.

U drugom poglavlju proširujemo Newtonovu metodu na glatke vektorske funkcije. Opet navodimo metodu i proučavamo teoreme konvergencije, s tim da se detaljnije bavimo lokalnom konvergencijom, dok globalnu konvergenciju samo ukratko opisujemo (iskazujemo jedan važan teorem i opisujemo jednu strategiju kako postići globalnu konvergenciju). Dalje opisujemo kako Newtonovom metodom možemo riješiti problem bezuvjetne optimizacije. Za kraj, implementiramo ovu metodu u Pythonu i isprobavamo implementaciju na konkretnom primjeru.

U trećem poglavlju proučavamo neglatke funkcije. Uvodimo pojmove generaliziranog B-subdiferencijala i (Clarkeovog) generaliziranog diferencijala. Pomoću generaliziranog diferencijala uvodimo Newtonovu metodu za neglatke sustave. Poslije toga uvodimo pojam poluglatkih funkcija, što nam je važan pojam pri proučavanju teorema konvergencije Newtonove metode za neglatke sustave (koju nazivamo poluglatkom Newtonovom metodom kada ju primijenjujemo na poluglatke funkcije). Poslije ukratko opisujemo kako Newtonovu metodu možemo proširiti na proizvoljne Banachove prostore.

Također pokazujemo kako se Newtonove metoda može iskoristiti da efikasno numerički riješimo ODJ. Dano je nekoliko primjera u kojima rješavamo ODJ prvog i drugog reda.

Summary

In this paper, we observe Newton's method and its various generalizations.

In the first chapter, we observe the standard Newton's method for smooth real-valued functions of a real variable. We motivate the method, describe it, and present the convergence theorems and error estimates. We briefly describe how the situation changes when we have multiple zeros. Finally, we implement this method in Python and give few more examples.

In the second chapter, we extend Newton's method so that it can be applied to smooth vector functions. Again, we describe the method and study the convergence theorems, dealing in more detail with local convergence, while we only briefly describe global convergence (we state one important theorem and describe one strategy on how to achieve global convergence). We further describe how the Newton method can be used to solve the problem of unconstrained optimization. Finally, we implement this method in Python and test the implementation on a concrete example.

In the third chapter, we study nonsmooth functions. We introduce the concepts of generalized B-subdifferential and (Clarke's) generalized differential. Using the generalized differential, we introduce Newton's method for nonsmooth systems. We then introduce the notion of semismooth functions, which is an important term for us when studying the convergence theorems of Newton's method for nonsmooth systems (which we call the semismooth Newton's method when we apply it to semismooth functions). We then briefly describe how we can extend Newton's method to arbitrary Banach spaces.

We also show how Newton's method can be used to efficiently numerically solve ODEs. Several examples are given in which we solve first- and second-order ODEs.

Životopis

Dana 25. studenog 1996. rođen sam u Karlovcu, gdje sam pohađao Osnovnu školu Dragojle Jarnević i Gimnaziju Karlovac. Već pri upisu u srednju školu sam se opredijelio za prirodoslovne predmete, pa sam pohađao prirodoslovno-matematičku gimnaziju.

Godine 2015. sam upisao Preddiplomski sveučilišni studij Matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu, koji sam 2018. godine i završio. Tada upisujem Diplomski sveučilišni studij Računarstva i matematike te Diplomski sveučilišni studij Financijske i poslovne matematike, oba na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu.