2. naloga: Iskanje ničel funkcij in numerična integracija v 1D

Gašper Harej

11. marec 2025

1 Uvod

V tem poročilu so predstavljene rešitve naloge, ki obravnava iskanje ničel polinomskih in drugih funkcij ter numerično integracijo v eni dimenziji. Uporabljene so klasične metode (metoda bisekcije, regula falsi) in naprednejši algoritmi (Brentova metoda, funkcije iz knjižnic NumPy in SciPy). V nadaljevanju so prikazani tudi primeri grafov, ki prikazujejo natančnost integracijskih metod, odklon dobljenih ničel od analitičnih vrednosti in primerjavo hitrosti konvergence različnih pristopov. Vključeni so tudi časovni primerjalni grafi, ki ponazarjajo učinkovitost metod.

2 Iskanje ničel funkcij

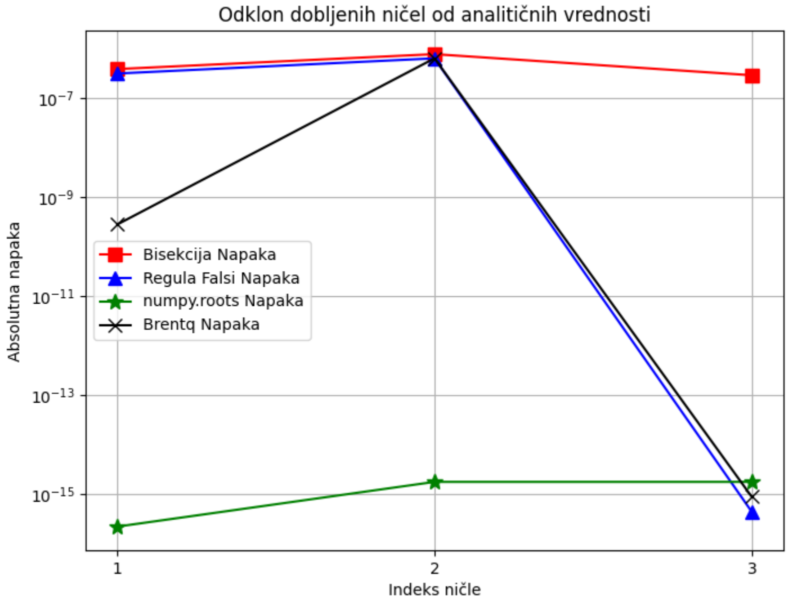
2.1 Uporabljene metode

- **Bisekcija:** preprosta, robustna, vendar le linearna konvergenca.
- **Regula falsi:** podobna bisekciji, a z uporabo linearne interpolacije za določitev nove točke.
- **Brent:** kombinira robustnost in hitrost (na voljo v `scipy.optimize.brentq`).
- **numpy.roots:** namensko razreševanje polinomov, zelo hitro in natančno za nižje stopnje.
- **Cardanovo:** analitična formula za polinome 3. stopnje.

2.2 Analitične rešitve in primerjava

Za polinome 3. stopnje lahko analitične ničle izračunamo s Cardanovo formulo. Tako dobimo neposredno primerjavo med dobljenimi numeričnimi rešitvami in dejanskimi vrednostmi ničel, kar omogoča natančno oceno relativne napake.

2.3 Odklon dobljenih ničel od analitičnih vrednosti (polinom $x^3 - 6x^2 + 11x - 6$)



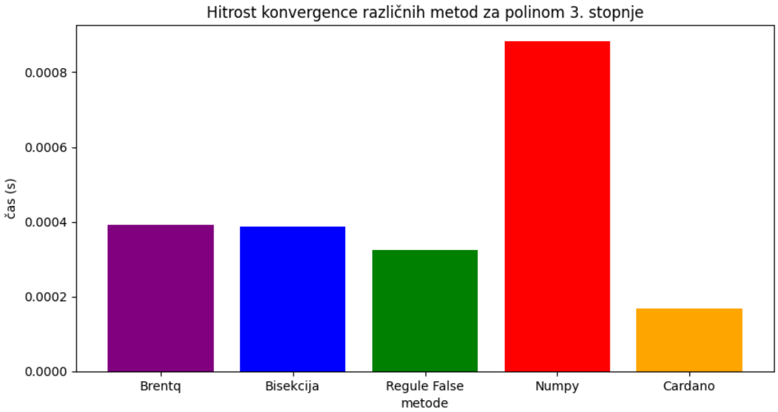
Slika 1: Odklon dobljenih ničel (absolutna napaka) od analitičnih vrednosti za različne metode pri polinomu $x^3 - 6x^2 + 11x - 6$.

Na sliki 1 so prikazani rezultati primerjave ničel polinoma $x^3 - 6x^2 + 11x - 6$, kjer so analitične vrednosti dobljene s Cardanovo formulo. Vidimo, da:

- **Bisekcija (modra)** in **Regula Falsi (rdeča)** dosegata približno podobno natančnost.
- **numpy.roots (zelena)** je izredno natančna, saj gre za namensko funkcijo za reševanje polinomov.
- **Brentova metoda (črna)** prav tako dosega zelo dobro natančnost (tudi do pod 10^{-13}).

Pri večini metod je natančnost okrog 10^{-7}, kar je v praksi več kot dovolj za številne aplikacije.

2.4 Hitrost konvergence metod pri iskanju ničel (polinom $x^3 - 6x^2 + 11x - 6$)



Slika 2: Hitrost konvergence (čas izvedbe) različnih metod pri polinomu $x^3 - 6x^2 + 11x - 6$.

Slika 2 prikazuje čas izvedbe (v sekundah) za več metod pri istem polinomu:

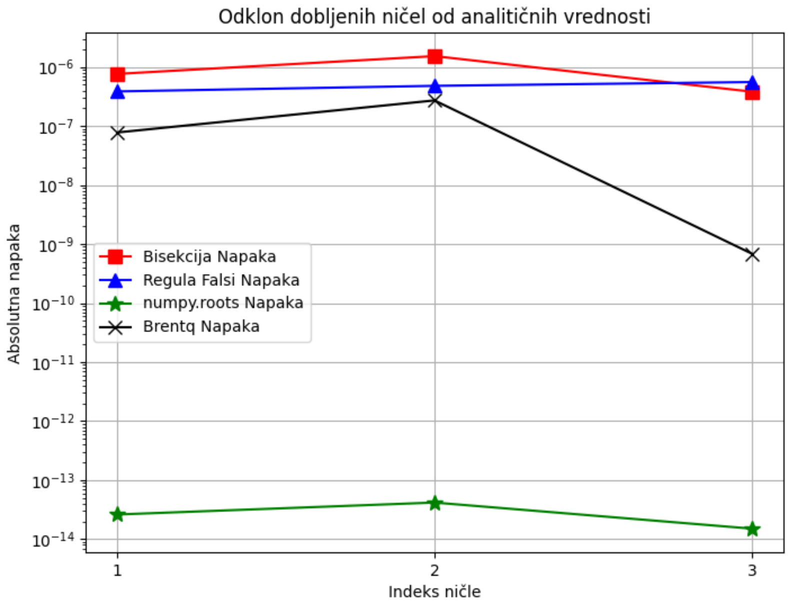
- **Brentova metoda (vijolična)** in **Regula Falsi (zelena)** sta konkurenčni glede hitrosti.
- **Bisekcija (modra)** je robustna, a praviloma počasnejša.
- **NumPy (rdeča)** pričakoval sem, da bo ta metoda hitrejša.
- **Cardanovo (oranžna)** je analitična formula, ki je za 3. stopnjo zelo učinkovita.

2.5 Rezultati za polinom $x^3 - 8x^2 + 20.75x - 17.5$

Polinom

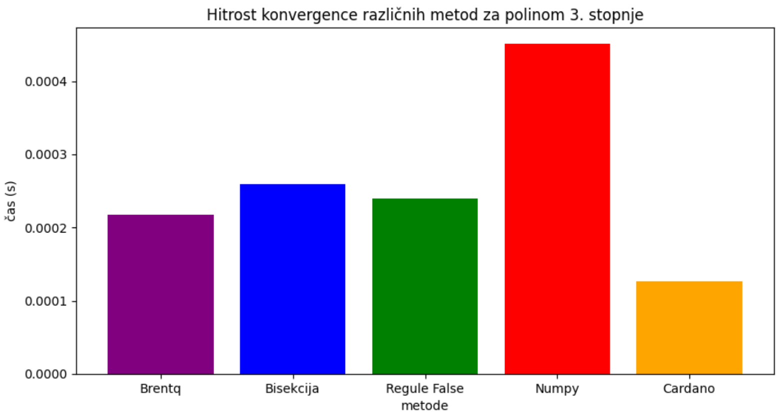
$$x^3 - 8x^2 + 20.75x - 17.5$$

ima ničle 2, 2.5 in 3.5. Spodaj sta prikazana grafa, ki sta analogna prejšnjima, vendar tokrat za ta polinom.



Slika 3: Odklon dobljenih ničel (absolutna napaka) od analitičnih vrednosti za različne metode pri polinomu $x^3 - 8x^2 + 20.75x - 17.5$.

Na sliki 3 je razvidno, da so vse metode dosegale zelo majhne napake (pod 10^{-6}), pri čemer sta **Brentova metoda** in **numpy.roots** znova dosegli odlične rezultate, podobno kot pri polinomu $x^3 - 6x^2 + 11x - 6$.



Slika 4: Hitrost konvergence (čas izvedbe) različnih metod pri polinomu $x^3 - 8x^2 + 20.75x - 17.5$.

Tudi pri časovni primerjavi (slika 4) se potrjuje, da sta **Brentova metoda** in **Cardano** med najhitrejšimi, medtem ko sta **Bisekcija** in **Regula Falsi** nekoliko počasnejši, a še vedno dovolj učinkoviti za večino praktičnih primerov.

3 Numerična integracija

3.1 Uporabljene metode

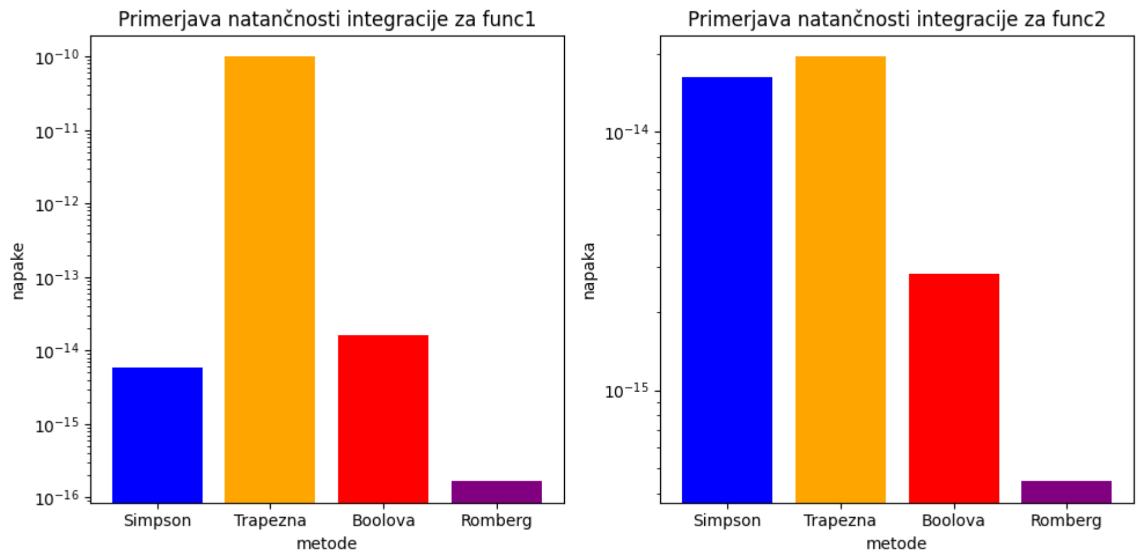
- **Trapezna metoda:** enostavna implementacija, napaka reda $O(h^2)$.
- **Simpsonova metoda:** natančnejša interpolacija, napaka reda $O(h^4)$.

- **Boolova (Bodejeva) metoda:** metoda višjega reda, ki lahko pri gladkih funkcijah še izboljša natančnost.
- **Rombergova metoda:** kombinira Richardsonovo ekstrapolacijo s trapezno metodo za hitro izboljševanje natančnosti.

3.2 Funkcije za testiranje integracije

- **func1:** $(1 - x^2)$ na intervalu $[-1, 1]$, kjer je analitična vrednost integrala $\frac{4}{3}$.
- **func2:** $(x^3 e^{-x})$ na intervalu $[0, \infty)$, pri čemer za praktično implementacijo določimo zgornjo mejo (npr. $x = 40$). Analitična vrednost integrala je $3! = 6$ (po definiciji gama funkcije $\Gamma(n + 1) = n!$).

3.3 Natančnost integracije

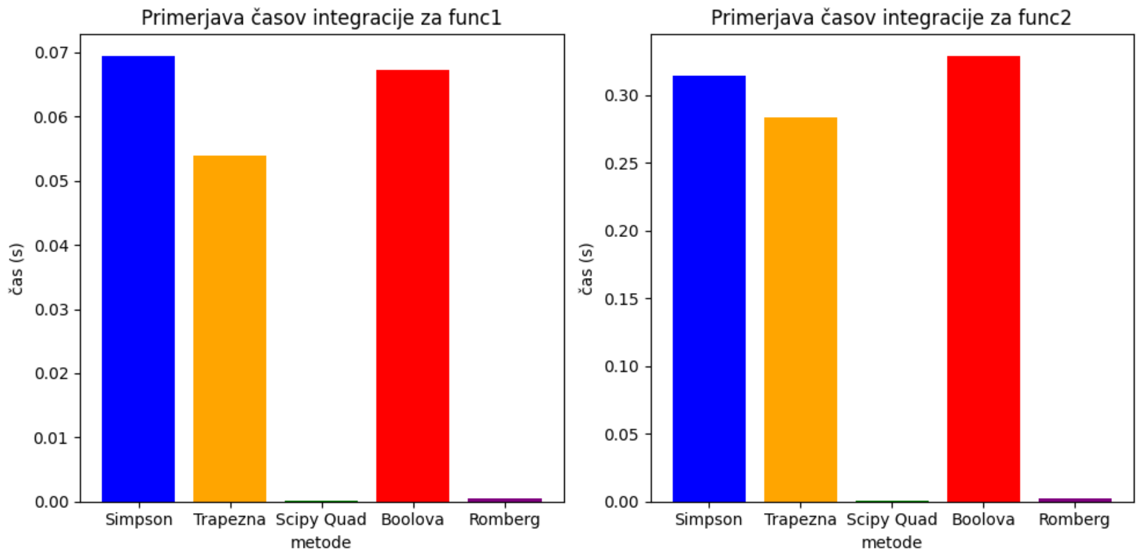


Slika 5: Primerjava natančnosti integracije za `func1` (levo) in `func2` (desno). Prikazane so logaritemske vrednosti napake za Simpsonovo, Trapezno, Boolovo in Rombergovo metodo.

Na sliki 5 je prikazana primerjava napak (vertikalna os je v logaritemski skali) za različne integracijske metode pri dveh funkcijah:

- **func1:** (levo) Boolova (rdeča) in Simpsonova metoda (modra) imata opazno manjšo napako kot trapezna (oranžna). Rombergova (vijolična) doseže še manjše napake.
- **func2:** (desno) Podobna slika, vendar so razlike napak nekoliko večje zaradi hitrega padanja funkcije pri večjih x. Kljub temu Boolova in Rombergova metoda dosegata napake pod 10^{-14}.

3.4 Časovna primerjava integracijskih metod



Slika 6: Primerjava časov integracije za `func1` (levo) in `func2` (desno). Metode Simpson, Trapezna, SciPy Quad, Boolova in Romberg.

Na sliki 6 sta prikazana dva stolpčna diagrama, ki primerjata čas (v sekundah) za izračun integrala z različnimi metodami:

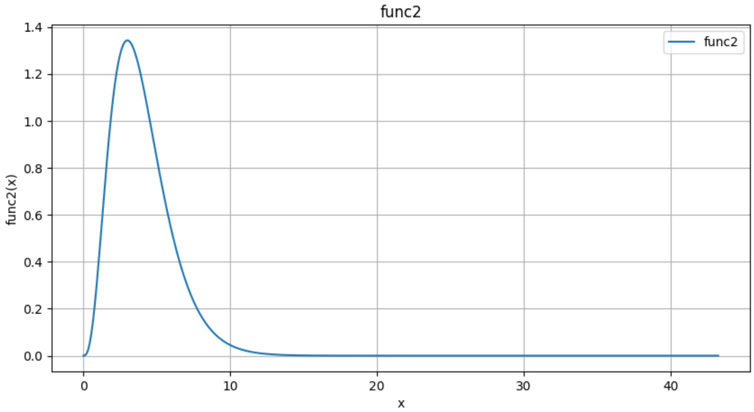
- **Simpsonova in Trapezna** sta praviloma zelo hitri, saj sta implementacijsko preprosti.
- **SciPy Quad** uporablja adaptivni algoritem (npr. QAGS), kar je lahko hitrejše ali počasnejše, odvisno od funkcije.
- **Boolova in Rombergova** porabita nekoliko več časa za posamezen izračun, a lahko dosežeta izredno natančnost z manj koraki.

  V mojem primeru sta tako **Rombergova** kot **SciPy Quad** izjemno hitrajši od drugih, ki so si po času primerljive.

4 Zaključek

Pri računanju integrala funkcije (x^3e^{-x}) na intervalu $[0, \infty)$ je bilo neskočnost pravilno oceniti:

4.1 Obnašanje funkcije `func2`



Slika 7: Graf funkcije `func2` (npr. $x^3 e^{-x}$) v razponu $x \in [0, 40]$.

Slika 7 prikazuje potek funkcije `func2`, ki ima največjo vrednost v območju okrog $x \approx 3$, nato pa hitro pada proti nič.

Iz slike je razvidno, da bi najverjetneje zadoščal že $x \approx 15$, vendar se tega nisem želel lotiti kar po občutku, zato sem spisal naslednjo funkcijo:

```
    def neskoncnost(f, r, korak, eps):
    i = 1.0
    while i < r:
        d = abs(f(i) - 0)
        if d < eps:
            return i
        i += korak
    return int(r)

neskonc = neskoncnost(func2, 1000, 0.01, 1e-6)
```

Funkcija išče, kdaj je vrednost primerljiva z nič in mi je ob $eps = 10^{-9}$ vrnila vredonst $x \approx 40$. Problem se je pojavil, ker je `Rombergova` metoda pri teh vrednostih v repu funkcije imela velike napake, kar sem odpravil z zmanjšanjem epsilona, tako da sem na koncu integriral do $x \approx 20$.