Matematička analiza - projektni zadatak

Denis Beletić

27. lipnja 2025.

Sadržaj

1 Uvod

Taylorov red predstavlja važan alat u matematičkoj analizi jer omogućava približno izražavanje složenih funkcija pomoću polinoma. Ovime se olakšavaju računanja i proučavanje svojstava funkcija u okolini određene točke. Omogućava preciznu analizu pogreške aproksimacije, što čini važan temelj teorije pogrešaka u matematici.

Na GitHub-u se nalazi repozitorij unutar kojeg možete preuzeti sve datoteke vezane za ovaj projektni zadatak: `https://github.com/denisbeletic/MAT_ProjektniZadatak`

2 Taylorov i Maclaurinov red

Taylorov red:

$$T_n(x) = f(c) + \frac{f'(c)}{1!}(x-c) + \frac{f''(c)}{2!}(x-c)^2 + \cdots + \frac{f^{(n)}(c)}{n!}(x-c)^n$$

Maclaurinov red (poseban slučaj kada je c = 0):

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n = f(0) + \frac{f'(0)}{1!} x + \frac{f''(0)}{2!} x^2 + \cdots$$

3 Zadatak 1.

Odredite Taylorov polinom funkcije

$$f(x) = \frac{1}{1+x^2}$$

u točki $x = 0$, koji sadrži prva tri člana koji su različiti od nule.
Pokažite sve korake deriviranja i uvrštavanja.
Napomena: Može biti jednostavnije naći Taylorov polinom za funkciju $\frac{1}{1+x}$, i zatim u tu formulu uvrstiti x^2.

Pošto tražimo Taylorov polinom u točki x = 0, možemo koristiti **Maclaurinov red** za pronalazak polinona.

Korak 1: Odaberimo jednostavniju funkciju

Promatramo jednostavniju funkciju:

$$f(x) = \frac{1}{1+x} = (1+x)^{-1}$$

Korak 2: Derivirajmo funkciju

$$\begin{aligned}
f(x) &= (1+x)^{-1} \\
f'(x) &= -1 \cdot (1+x)^{-1-1} = -1 \cdot (1+x)^{-2} = -(1+x)^{-2} \\
f''(x) &= -1 \cdot (-2) \cdot (1+x)^{-2-1} = 2 \cdot (1+x)^{-3} \\
f'''(x) &= 2 \cdot (-3) \cdot (1+x)^{-3-1} = -6 \cdot (1+x)^{-4}
\end{aligned}$$

Korak 3: Vrijednosti derivacija u točki $x = 0$

$$\begin{aligned}
f(0) &= (1+0)^{-1} = 1 \\
f'(0) &= -1 \\
f''(0) &= 2 \\
f'''(0) &= -6
\end{aligned}$$

Korak 4: Maclaurinov red za $(1+x)^{-1}$

Uvrštavanjem u formulu:

$$\begin{aligned}
f(x) &= f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \cdots \\
f(x) &= 1 + (-1)x + \frac{2}{2}x^2 + \frac{-6}{6}x^3 + \cdots \\
f(x) &= 1 - x^1 + x^2 - x^3 + \cdots
\end{aligned}$$

Korak 5: Zamjena $x \to x^2$

Sada zamjenjujemo x s x^2:

$$(1 + x^2)^{-1} = 1 - (x^2)^1 + (x^2)^2 - (x^2)^3 + \cdots$$

Rezultat:

Taylorov polinom s prva tri nenulta člana:

$$P(x) = 1 - x^2 + x^4 - x^6$$

4 Zadatak 2.

Napišite kod u R-u ili Pythonu koji vizualizira funkciju $f(x)$ i Taylorove polinome s jednim, dva i tri člana na segmentu $[-0.8, 0.8]$.
Uočite da konkretne Taylorove polinome koje koristite trebate izračunati u prvom dijelu zadatka i da u ovom dijelu ne pišete da koristite $T_1(x)$, $T_2(x)$ i $T_3(x)$, već one polinome koji koriste članove razvoja koji su različiti od nule (i tako do trećeg nenultog člana).

Python kod za vizualizaciju

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

def f(x):
    return 1 / (1 + x**2)

def p1(x):
    return 1 - x**2

def p2(x):
    return 1 - x**2 + x**4

def p3(x):
    return 1 - x**2 + x**4 - x**6

x = np.linspace(-0.8, 0.8, 400)
# generira 400 brojeva, medusobno ravnomjerno rasporedeni u intervalu [-0.8, +0.8]

plt.figure(figsize=(8,6))
plt.plot(x, f(x), label='f(x) = 1 / (1 + x^2)', color='black')
plt.plot(x, p1(x), label='Taylorov 1. član')
plt.plot(x, p2(x), label='Taylorov 2. član')
plt.plot(x, p3(x), label='Taylorov 3. član')
plt.legend()
plt.title('Vizualizacija fj-e i Taylorovi polinomi')
plt.xlabel('x')
plt.ylabel('y')
plt.grid(True)
plt.show()
```

Listing 1: Vizualizacija funkcije i Taylorovih polinoma

5 Zadatak 3.

Kvantificirajte točnost aproksimacije Taylorovim polinomima tako da za svaki Taylorov polinom $T_n(x)$, gdje je n jednak onim stupnjevima polinoma određenima u točkama 1 i 2, numerički izračunate:

$$\max_{x \in [-0.8, 0.8]} |f(x) - T_n(x)|$$

Napomena: Ovdje nije potrebna analitička procjena greške – dovoljno je numerički usporediti funkcijske vrijednosti u velikom broju točaka (recimo 1000).

Python kod za kvantifikaciju točnosti aproksimacije

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

def f(x):
    return 1 / (1 + x**2)

def p1(x):
    return 1 - x**2

def p2(x):
    return 1 - x**2 + x**4

def p3(x):
    return 1 - x**2 + x**4 - x**6

# MAX, x in [-0.8, +0.8]  |  f(x) - Tn(x) |
x_vrijednosti = np.linspace(-0.8, 0.8, 1000)
greske_p1 = np.abs(f(x_vrijednosti) - p1(x_vrijednosti))
greske_p2 = np.abs(f(x_vrijednosti) - p2(x_vrijednosti))
greske_p3 = np.abs(f(x_vrijednosti) - p3(x_vrijednosti))

# f(), p1(), p2() i p3() s argumentom 'x_vrijednosti' vraća niz rezultata jer je 'x_vrijednosti' numpy array
# rezultat ovog gornjih računa je niz razlika, odnosno grešaka za svaku od 1000 točaka

najveca_greska_p1 = np.max(greske_p1)
najveca_greska_p2 = np.max(greske_p2)
najveca_greska_p3 = np.max(greske_p3)

print(f"Najveća greška za p1: {najveca_greska_p1}")
print(f"Najveća greška za p2: {najveca_greska_p2}")
print(f"Najveća greška za p3: {najveca_greska_p3}")
```

Listing 2: Kvantifikacija točnosti aproksimacije Taylorovih polinoma

6 Zadatak 4.

Prikažite dobivene pogreške u tablici i komentirajte što se događa s pogreškom kada se red produljuje. Kada Taylorova aproksimacija dobro funkcionira, a kada gubi točnost?

Pokretanjem gore navedenog koda, dobiti ćemo slj. max. greške:

Broj nenultih članova	**Max. greška na** $[-0{,}8,\ 0{,}8]$
1 (polinom p_1)	0.24975609756097572
2 (polinom p_2)	0.15984390243902435
3 (polinom p_3)	0.10230009756097569

Iz tablice vidimo da povećanjem broja nenultih članova Taylorovog polinoma, max. se pogreška smanjuje. To znači da, što više članova polinoma iskoristimo, aproksimacija postaje sve točnija i točnija.

Taylorova aproksimacija funkcionira sve bolje i bolje kako se povećava broj članova polinoma - odnosno, što je red polinoma veći.