Univerzitet u Sarajevu
Elektrotehnički fakultet
Predmet: Optimizacija resursa
Akademska godina: 2020/2021
Student: Samra Mujčinović
Datum: 3.11.2020.

Izvještaj sa laboratorijske vježbe 2

Zadatak 1

Ideja za rješenje zadatka

Za realizaciju ovog zadatka prije svega bilo je potrebno upoznati se sa Newton-Raphsonovim algoritmom za traženje lokalnog ekstrema funkcije f(x). Za implementaciju ovog algoritma iskoristili smo odgovarajuću formulu, te nakon što algoritam završi vrati se vrijednost lokalnog ekstrema.

Matematski opis zadatka

Za implementaciju ovog algoritma korištena je rekurzivna relacija Newton-Raphsonovog algoritma za nalaženje lokalnog ekstrema funkcije f(x):

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \tag{1}$$

Uslov zaustavljanja ovog algoritma je dostizanje maksimalnog broja iteracija N ili kada bude ispunjeno:

$$|f(x_{k+1}) - f(x_k)| < epsilon \tag{2}$$

Nakon čega se usvaja da je x_{k+1} rješenje, tj tačka lokalnog ekstrema.

Vodič kroz kôd/simulaciju

U kodu je definisano nekoliko funkcija. Prva od njih je funkcija koja računa prvi izvod date funkcije u nekoj zadanoj tački, a odmah nakon nje je definisana i druga funkcija za računanje drugog izvoda sa istim parametrima kao prethodna. Obje funkcije vraćaju numeričku vijednost izvoda funkcije u zadanoj tački.

```python
def PrviIzvod(f, x):
    return (f(x + 0.001) - f(x)) / 0.001


def DrugiIzvod(f, x):
    return (f(x + 0.001) - 2*f(x) + f(x - 0.001)) / (0.001)**2
```

Slika 1: Funkcije za računanje prvog i drugog izvoda

Sljedeća funkcija objedinjuje implementaciju Newton-Raphsonovog algoritma za traženje ekstrema zadane funkcije i u nastavku je dat njen prototip:

```python
def NewtonRaphsonAlgoritm(f, df, ddf, x0, n, eps)
```

Parametri ove funkcije su:

- f – funkcija za koju tražimo ekstreme
- df – prvi izvod date funkcije f u zadanoj tački x0
- ddf – drugi izvod date funkcije f u zadanoj tački x0
- x0 – proizvoljna početna tačka
- n - maksimalan broj iteracija
- eps – minimalna promjena po vrijednosti funkcije između dvije sukcesivne iteracije

Unutar ove funkcije definisana je jedna “for” petlja u kojoj je sadržana logika ovog algoritma – definisani su uslovi zaustavljanja algoritma, te je iskorištena prethodno navedena formula kojom se računa ekstrem funkcije. Konačno, ova funkcija vraća numeričku vrijednost ekstrema u zadanoj tački.

Definisane su još 4 funkcije za provjeru ispravnosti rada algoritma koje su identične onima koje su zadane u drugom zadatku.

Rezultati kôda/simulacije

U ovom dijelu izvještaja prikazat ćemo rezultate ovog algoritma za različite funkcije i vrijednosti parametara:

$N=10, eps=0$

1. $f(x)=3x^2-1, x_0\in\{1,10\}$

```
samra@samra-laptop:~$ /usr/bin/python3 "/home/samra/Documents/Optimizacija resursa/lab2/zadatak1.py"
Rezultat izvršavanja prve funkcije za x0 = 1: -0.0005000000000000929
Rezultat izvršavanja prve funkcije za x0 = 10: -0.0005000000000000195
```

2. $f(x)=-(16x^2-24x+5)e^{-x}, x_0\in\{0.5,2\}$

```
Rezultat izvršavanja druge funkcije za x0 = 0.5: 0.6314661318512653
Rezultat izvršavanja druge funkcije za x0 = 2: 2.8675340348155625
```

3. $f(x)=\sin(x)+\sin(\frac{10}{3}x), x_0 \in \{3,6,7\}$

```
Rezultat izvršavanja trece funkcije za x0 = 3: 3.3867517560267415
Rezultat izvršavanja trece funkcije za x0 = 6: 6.216808890325152
Rezultat izvršavanja trece funkcije za x0 = 7: 6.999649085623919
```

4. $f(x)=e^{-x}, x_0 \in \{1,10\}$

```
Rezultat izvršavanja cetvrte funkcije za x0 = 1: 10.99500083323269
Rezultat izvršavanja cetvrte funkcije za x0 = 10: 19.99500083384769
```

$N=50, eps=0$

```
samra@samra-laptop:~$ /usr/bin/python3 "/home/samra/Documents/Optimizacija resursa/lab2/zadatak1.py"
Rezultat izvršavanja prve funkcije za x0 = 1: -0.0005000000000000929
Rezultat izvršavanja prve funkcije za x0 = 10: -0.0005000000000000195
Rezultat izvršavanja druge funkcije za x0 = 0.5: 0.6314661318512653
Rezultat izvršavanja druge funkcije za x0 = 2: 2.8675340348155625
Rezultat izvršavanja trece funkcije za x0 = 3: 3.3867517560267415
Rezultat izvršavanja trece funkcije za x0 = 6: 6.216808890325152
Rezultat izvršavanja trece funkcije za x0 = 7: 6.999649085623919
Rezultat izvršavanja cetvrte funkcije za x0 = 1: 50.975004166136884
Rezultat izvršavanja cetvrte funkcije za x0 = 10: 59.9750041678121
```

$N=100, eps=0.001$

```
samra@samra-laptop:~$ /usr/bin/python3 "/home/samra/Documents/Optimizacija resursa/lab2/zadatak1.py"
Rezultat izvršavanja prve funkcije za x0 = 1: -0.0005000000000000929
Rezultat izvršavanja prve funkcije za x0 = 10: -0.0005000000000000195
Rezultat izvršavanja druge funkcije za x0 = 0.5: 0.6314646221438899
Rezultat izvršavanja druge funkcije za x0 = 2: 2.867435423493635
Rezultat izvršavanja trece funkcije za x0 = 3: 3.3867421445365484
Rezultat izvršavanja trece funkcije za x0 = 6: 6.216808796989417
Rezultat izvršavanja trece funkcije za x0 = 7: 6.999649000097747
Rezultat izvršavanja cetvrte funkcije za x0 = 1: 7.996500583524919
Rezultat izvršavanja cetvrte funkcije za x0 = 10: 10.999500083282715
```

Zaključak

Na osnovu izvršenih provjera možemo zaključiti da naš algoritam ispravno radi. Na prethodnim primjerima moguće je uočiti da u nekim slučajevima je moguća promjena vrijednosti ekstrema kako se mijenja broj iteracija i minimalna promjena po vrijednosti funkcije između dvije sukcesivne iteracije, te vrijednost izabrane tačke u kojoj se računa ekstrem funkcije, što će biti grafički prikazano u drugom zadatku.

Zadatak 2

Ideja za rješenje zadatka

Za realizaciju ovog zadatka iskoristit ćemo prethodno implementiran algoritam za pronalaženje ekstrema zadanih funkcija. Za grafički prikaz ovih funkcija, te obilježavanje nađenih ekstrema i rezultata implementiranog algoritma, iskoristit ćemo pogodnosti Python programskog jezika i njegovog modula “matplotlib”.

Matematski opis zadatka

Matematički opis za ovaj zadatak je dat u opisu zadatka 1, jer koristi algoritam iz prethodnog zadatka.

Vodič kroz kôd/simulaciju

Za ovaj zadatak definisane su dvije nove funkcije, pri čemu prva služi za crtanje koordinatnih osa za svaku funkciju u odgovarajućem opsegu tako da funkcije budu što bolje prikazane i vrijednosti jasno predstavljene. Ova funkcija ima sljedeći prototip:

```
def NacrtajOse(x, y, xorg1 = -5, xorg2 = 50, yorg1 = -5, yorg2 = 20)
```

Kao parametre ova funkcija prima dva vektora x i y, te vrijednosti po x i y osi do kojih će grafik biti iscrtan, tj do koje maksimalne vrijednosti x i y će se grafik biti vidljiv. U prototipu su date predefinisane vrijednosti u slučaju da ih korisnik sam ne definise. Unutar ove funkcije objedinjena je logika iscrtavanja osa po zadanim parametrima.

Druga funkcija služi za crtanje grafika svake od zadanih funkcija i njen prototip je sljedeći:

```
def NacrtajGrafik1(f, naziv, n, eps, df, ddf)
```

Kao parametre ova funkcija prima:

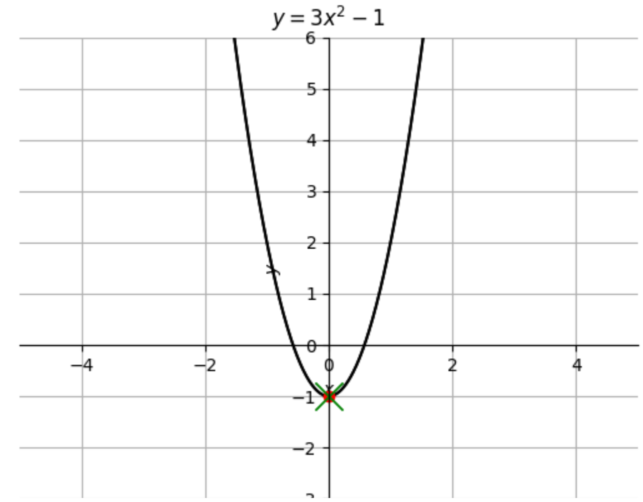
1. f – funkcija koja se crta
2. naziv – definicija funkcije koja se prikazuje na grafiku
3. n – maksimalan broj iteracija
4. eps - minimalna promjena po vrijednosti funkcije između dvije sukcesivne iteracije
5. df – prvi izvod funkcije u zadanoj tački
6. ddf – drugi izvod funkcije u zadanoj tački

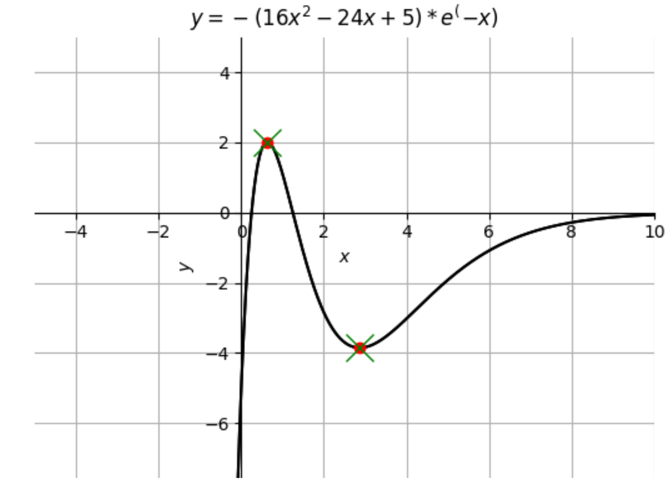
Unutar ove funkcije poziva se funkcija koja crta ose(prethodno objašnjena), a zatim se iscrtava odgovarajuća funkcija I označavaju ekstremi I rješenja algoritma. I unutar ove funkcije poziva se funkcija iz prethodnog zadatka za računanje ekstrema zadane funkcije.

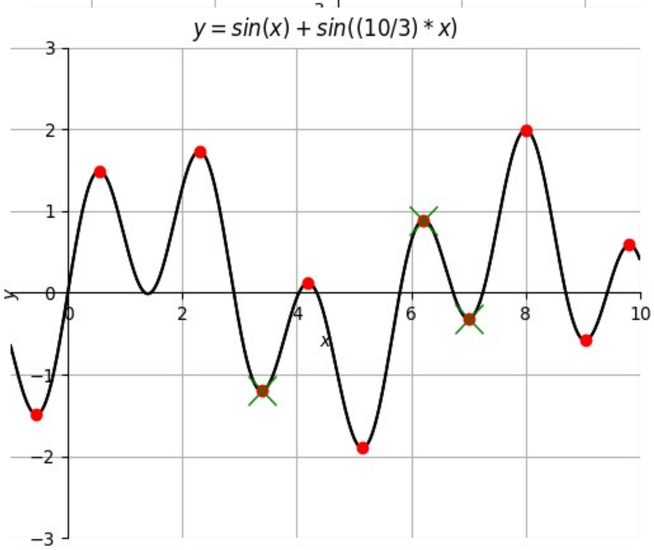
Rezultati kôda/simulacije

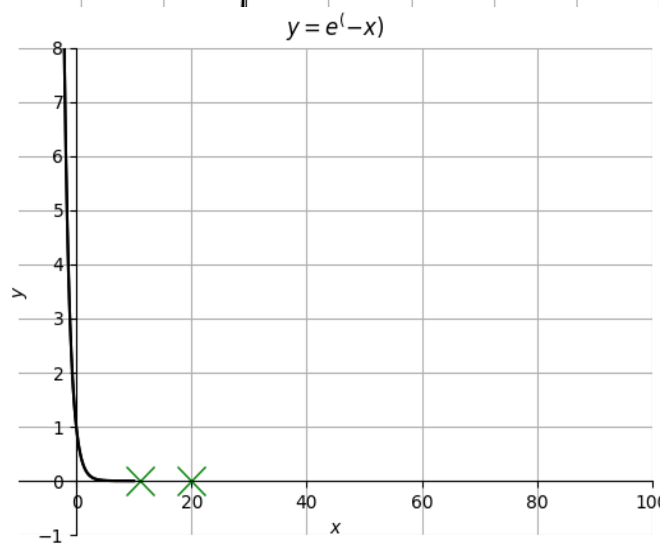
U prethodnom zadatku pokazali smo rezultate algoritma za različite vrijednosti parametara. U ovom zadatku iskoristit ćemo prethodne primjere za vrijednosti parametara i prikazati ih grafički, nakon čega ćemo izvući određene zaključke.

$N=10, eps=0$

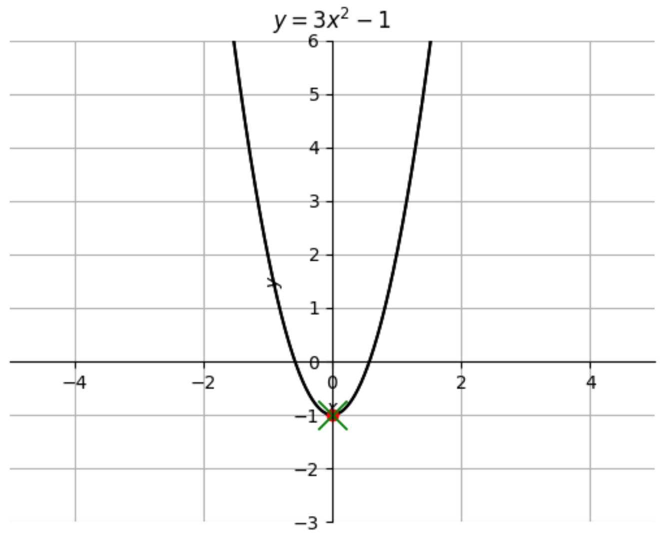


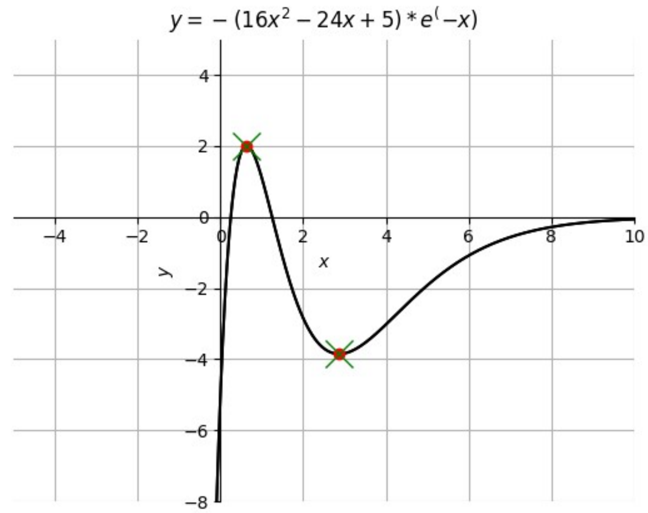


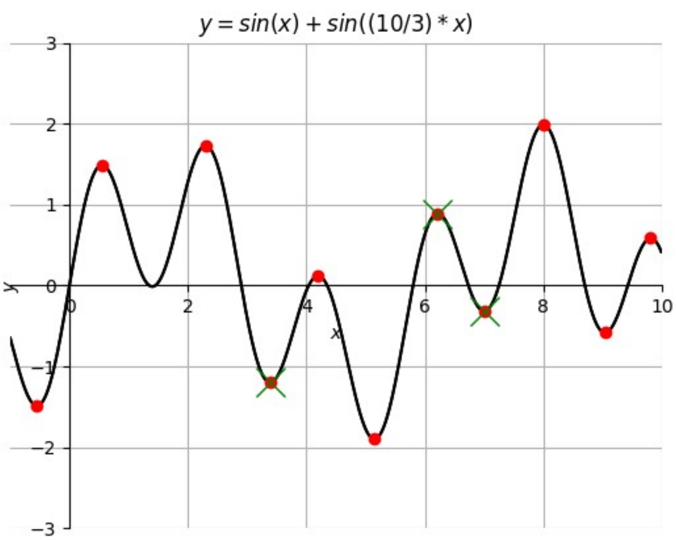


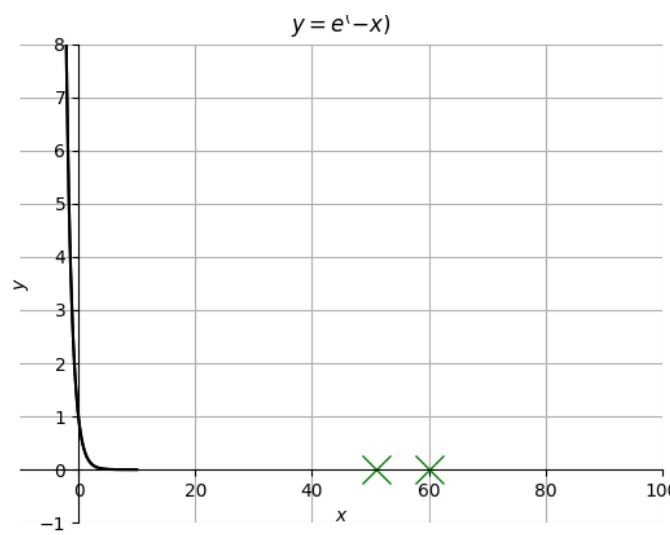


$N = 50, eps = 0$

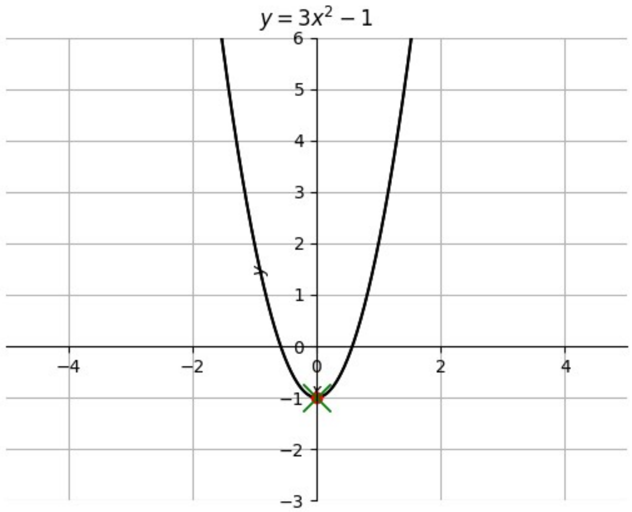


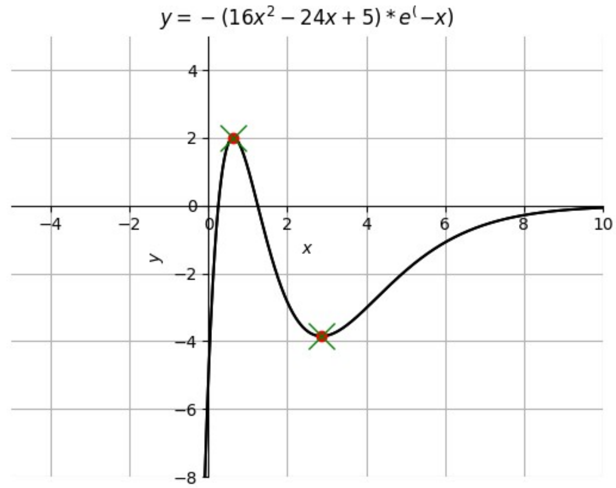


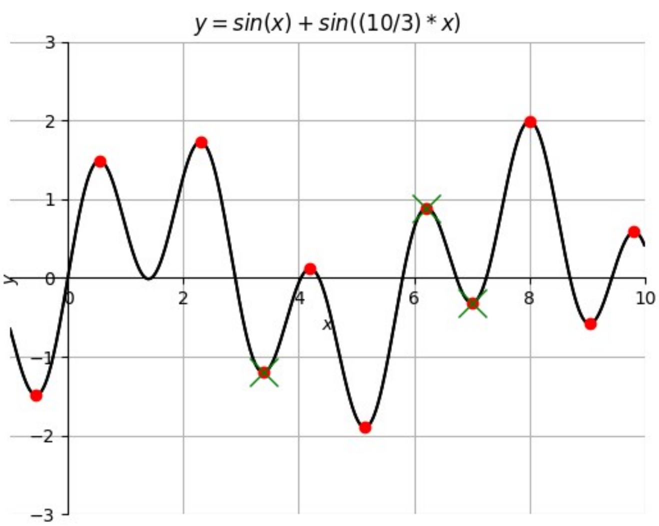


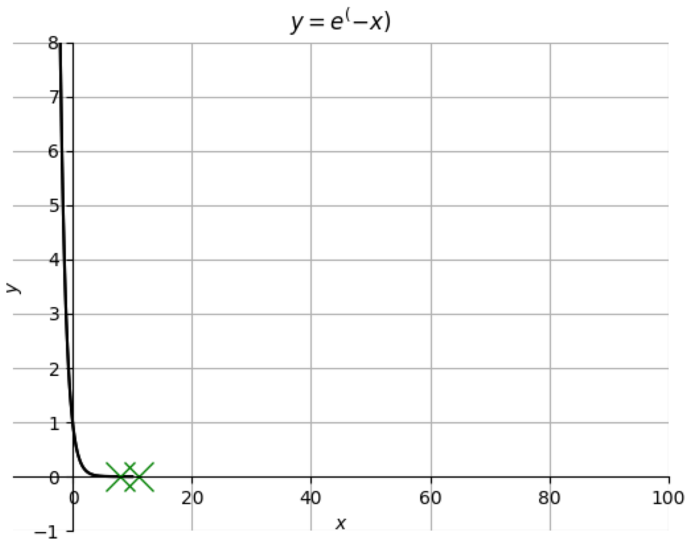


$N = 100\,, eps = 0.001$









Zaključak

Na osnovu datih primjera I njihovih grafika, najveću razliku možemo uočiti upravo na posljednjoj funkciji, kao što je to bio primjer u prethodnom zadatku kada smo vidjeli da se za različite vrijednosti dobijaju isti ili priblizni rezultati, dok su najveća odstupanja bila upravo na posljednjoj funkciji. Za drugačije vrijednosti proizvoljnih varijabli, moguće je I za druge funkcije dobiti mala odstupanja.