Amina Dacić

Prirodno-matematički fakultet

Teorijska kompjuterska nauka

5574/M

Program za određivanje maksimuma (minimuma) funkcije

KRATAK UVOD

Svrha dokumenta je opis metode za određivanje maksimuma (minimuma) zadane funkcije bez upotrebe derivacije i u drugom dijelu sa upotrebom derivacije i prikaz implemenacije algoritma za izračun te funkcije rađen u programskom jeziku Python. Prva metoda je metoda zlatnog reza.

METOD ZLATNOG REZA (GOLDEN SECTION METHOD/SEARCH)

Funkcije definisane na nekom intervalu u opštem slučaju mogu imati više od jednog maksimuma (minimuma), stim da samo jedan od njih nazivamo globalnim maksimumom (minimumom), dok su svi ostali lokalni maksimumi (minimumi).

Za bolje razumijevanje ovog metoda upoznaćemo se sa pojmom **unimodalne** funkcije.

Def: Za funkciju f**: [a, b]** → **R** kažemo da je unimodalna na intervalu **[a, b]** ako f postiže samo jedan maksimum (minimum).

Drugi zahtjev koji se postavlja funkcijama jeste njihova neprekidnost. Slijedi i definicija o neprekidnosti funkcije. Dakle:

Def: Za neku funkciju ***f*** kažemo da je neprekidna u nekoj tački c ako funkcija uopšte posjeduje vrijednost u toj tački, te ako se njena vrijednost kada se sa lijeva ili desna približavamo posmatranoj tački takođe približava njenoj vrijednosti u posmatranoj tački.

U ovom dokumentu su isključivo neprekidne unimodalne funkcije na zadanome intervalu **[a, b]**.

Ako je f(x) unimodalna funkcija na [a, b] tada je moguće zamijeniti taj interval sa podintervalom [a1, b1] na kojem f(x) dostiže svoj maksimum (minimum). Traženje maksimuma od f je ekvivalentno je traženju minimuma funkcije –f. Termin optimizacija ravnopravno se koristi s terminom minimizacija funkcije. Razmotrićemo slučaj pronalaženja minimuma, koji je analogan slučaju za maksimum.

Izaberu se dvije unutrašnje tačke iz intervala [a, b], c < d koje ostvaruju relaciju a < c < d < b. Uslov da je funkcija unimodalna obezbjeđuje da su vrijednosti f(c) i f(d) manje od max{f(a), f(b)}. Razmotrimo dva slučaja.

Ako je f(c) ≤ f(d) minimum se nalazi na podintervalu [a, d] pa mijenjamo b sa d i nastavljamo pretragu na novom intervalu [a, d].

Ako je f(d) < f(c) minimum se nalazi na podintervalu [c, b] pa mijenjamo a sa c i nastavljamo pretragu na novom intervalu [c, b].

Unutrašnje tačke c i d su izabrane tako da su rezultujući intervali [a, c] i [d, b] simetrični, tj. b – d = c – a, gdje je

$$c = a + (1 - r)(b - a) = a + (1 - r)b$$

$$d = b - (1 - r)(b - a) = (1 - r)a + b, \quad \frac{1}{2} < r < 1$$

Želimo da vrijednost od r ostane konstanta na svakom podintervalu. Jedna od starih unutrašnjih tačaka bit će iskorištena kao unutrašnja tačka novog podintervala, dok druga unutrašnja tačka će postati krajnja tačka novog podintervala. Zato se u svakoj narednoj iteraciji određuje samo jedna nova tačka i pravi samo jedna procjena vrijednosti funkcije.

Ako je f(c) ≤ f(d) i ako trebamo napraviti samo jednu pocjenu funkcije, tada moramo imati

$$\frac{d-a}{b-a}=\frac{c-a}{d-a}$$

$$\frac{r(b-a)}{b-a}=\frac{(1-r)(b-a)}{r(b-a)}$$

$$\frac{r}{1}=\frac{(1-r)}{r}=\text{¿}r^2+r-1=0=\text{¿}r=\frac{-1\pm\sqrt{5}}{2}$$

Kako je $\frac{1}{2}$ < r < 1, uzimamo da je $r=\frac{-1+\sqrt{5}}{2}$.

Primijetimo da broj r dijeli interval [a, b] u omjeru zlatnog presjeka i otuda potiče ime navedenog metoda. Analogno, ako je f(d) < f(c) tada ponovo dobijamo da je r = $\frac{-1+\sqrt{5}}{2}$.

Navedimo sada **algoritam** zlatnog reza.

1. Postaviti parametar tolerancije (prag tačnosti rješenja ε¿
2. Naći interval [a, b] koji sadrži traženi minimum (maksimum)
3. Ako je |a - b| < ε stani, rješenje je $\frac{a+b}{2}$
4. Izračunati f(a) i f(b)
5. Konstruisati dvije tačke c = a + $\frac{3-\sqrt{5}}{2}(b-a)$ i d = a + $\frac{\sqrt{5-1}}{2}(b-a)$
6. Izračunati f(c) i f(d)
7. Ako je f(c) > f(d) minimum se nalazi u [c, b], sada stavi a = c i vrati se na korak 3.
   Primijetimo da trenutno d postaje c u idućem koraku i time smanjujemo jedno izračunavanje f(x).
8. Ako je f(c) ≤ f(d) tada je minimum u [a, d], sada stavi d = b i vrati se na korak 3.
   Primijetimo da sada u idućem koraku trenutno c postaje d i smanjujemo jedno izračunavanje veličine f(x).

Slijedi i implementacija algoritma u Python-u.

KOD

```
import math
from math import e
```

```
broj1 = (math.sqrt(5) - 1) / 2
broj2 = (3 - math.sqrt(5)) / 2

def  zlatni_rez(f, a, b, tol = 1e-5, h = None, c = None, d = None, fc = None, fd = None):
    print(a, b, c, d, fc, fd)
    (a, b) = (min(a, b), max(a, b))
    if h is None: h = b - a
        if h <= tol:
            return (a, b)
    if c is None:  c = a + broj2 * h
    if d is None:  d = a + broj1 * h
    if fc is None:  fc = f(c)
    if fd is None: fd = f(d)
    if fc < fd:
        return zlatni_rez(f, a, d, tol, h*broj1, c = None, fc = None, d = c, fd = fc)
    else:
        return zlatni_rez(f, c, b, tol, h*broj1, c = d, fc = fd, d = None, fd = None)
```

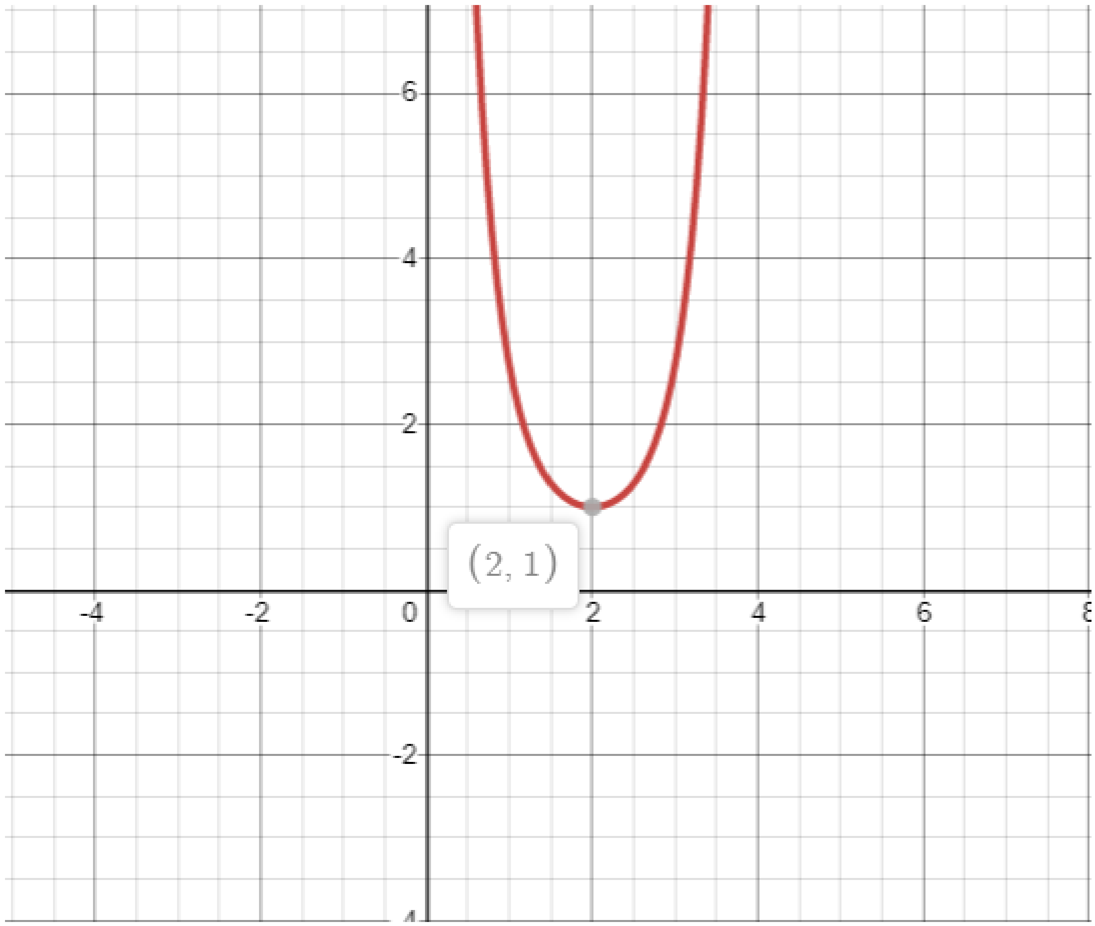
```
Glavni program:
```

```
# sada unos i ispis
```

```
f = lambda x: e**((x-2)**2)
a = -5
b = 5
(c, d) = zlatni_rez(f, a, b)
print(f((d + c)/2))
```

Dakle, program vraća 1.0 gdje data funkcija dostiže minimalnu vrijednost.

Grafički prikaz zadane funkcije:



Prikaz linije broj 7 odnosno prikaz rezultata pretrage za ekstremom.

-5	-5	None	None	None	None
-1.18033988	5	1.18033988	None	1.95784171	None
1.18033988	3.54101966	None	2.63932022	None	1.50490586
1.18033988	2.63932022	None	2.08203932	None	1.00675315
1.86917696	2.08203932	1.95048315	None	1.00245492	None
1.99340176	2.01259555	2.00073313	None	1.00000053	None
1.99997561	2.00001245	None	1.99999838	None	1.00000000
1.99998969	2.00001245	1.99999838	None	1.00000000	None
1.99998969	2.00000376	None	1.99999838	None	1.00000000
1.99999506	2.00000376	1.99999838	None	1.00000000	None

Sada ćemo iz drugog ugla posmatrati nalaženje maksimuma odnosno minimuma neke funkcije tj. upotrebom derivacije. Neka je funkcija ***f*** definisana u okolini tačke **a** i neka je **x** tačka iz te okoline. Ako koeficijent

$$\frac{f(x)-f(a)}{x-a}$$

ima graničnu vrijednost kada **x → a**, onda tu graničnu vrijednost zovemo derivacijom funkcije ***f*** u tački **a** i označavamo sa ***f*'(a)**. Dakle:

$$f\,'(a)=\lim_{x\to a}\frac{f(x)-f(a)}{x-a}$$

Navesti ćemo definicije za lokalni maksimum, lokalni minimum, kao i za strogi lokalni maksimum odnosno minimum:

- Funkcija ***f*** ima u tački **a strogi lokalni maksimum** ako postoji okolina tačke **a** tako da je ***f*(a) > *f*(x)** za svako **x** iz te okoline.
- Funkcija ***f*** ima u tački **a lokalni maksimum** ako postoji okolina tačke **a** tako da je ***f*(a) ≥ *f*(x)** za svako **x** iz te okoline.
- Funkcija ***f*** ima u tački **a strogi lokalni minimum** ako postoji okolina tačke **a** tako da je ***f*(a) < *f*(x)** za svako **x** iz te okoline.
- Funkcija ***f*** ima u tački **a lokalni minimum** ako postoji okolina tačke **a** tako da je ***f*(a) ≤ *f*(x)** za svako **x** iz te okoline.

Minimum i maksimum jednom riječju zovemo **ekstremima** funkcije ***f***.

POSTUPAK ODREĐIVANJA LOKALNIH EKSTREMA

Pretpostavimo da funkcija ***f*** ima derivaciju u svakoj tački područja definicije. Lokalne ekstreme funkcije ***f*** nalazimo ovim postupkom:

1. Odredimo stacionarne tačke funkcije ***f*** sa $f\,'(a)=0$
2. Odredimo drugu derivaciju i uvrstimo stacionarne tačke:
   - ako je broj < 0 imamo maksimum
   - ako je broj > 0 imamo minimum
3. Stacionarne tačke uvrstimo u početnu funkciju i dobijemo y

- Kratak primjer za ilustraciju gore navedenog:

$f(x)=2x^3+3x^2-36x-10$

1. Nađemo prvu derivaciju i izjednačimo ju sa 0:

$f'(x)=2\cdot3x^{3-1}+3\cdot2x^{2-1}-36\cdot1-0=6x^2+6x-36$

$6x^2+6x-36=0 \quad /:6$

$x^2+x-6=0$

Prema formuli $x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$ dobijamo da je:

$x_1=2$ i $x_2=-3$ → stacionarne tačke

2. Nađemo drugu derivaciju i uvrstimo stacionarne tačke da vidimo imamo li MAX ili min:

$f''(x)=6\cdot2x^{2-1}+6\cdot1-0=12x+6$

$f''(2)=12\cdot2+6=24+6=30>0$ minimum

$f''(-3)=12\cdot(-3)+6=-36+6=-30<0$ MAXIMUM

3. Stacionarne tačke uvrstimo u početnu funkciju da dobijemo y:

$f(2)=2\cdot2^3+3\cdot2^2-36\cdot2-10=16+12-72-10=-54$

$f(-3)=2\cdot(-3)^3+3\cdot(-3)^2-36\cdot(-3)-10=-54+27+108-10=71$

Te slijedi da je **min(2,-54)** i **MAX(-3,71).**

Dakle, do sada smo posmatrali nalaženje maksimuma i minimuma zadane funkcije kroz matematičke definicije i primjer. Sada je na redu implementacija algoritma.

KOD

```python
from sympy import *
x = Symbol('x', real=True)
f = 2*x**3 + 3*x**2 - 36*x - 10
prvi_izvod = f.diff(x)
l = solve(prvi_izvod, x)
x1 = l[0]
x2 = l[1]
drugi_izvod = f.diff(x).diff(x)
drugi_izvod_za_x1 = drugi_izvod.subs(x, x1)
drugi_izvod_za_x2 = drugi_izvod.subs(x, x2)

maksimum_za_x = 0
minimum_za_x = 0
if drugi_izvod_za_x1 < 0:
     maksimum_za_x = x1
     minimum_za_x = x2
elif drugi_izvod_za_x1 > 0:
     maksimum_za_x = x2
     minimum_za_x = x1
elif drugi_izvod_za_x2 < 0:
     maksimum_za_x = x2
     minimum_za_x = x1
elif drugi_izvod_za_x2 > 0:
     maksimu_za_x = x1
     minimum_za_x = x2

prvi_izvod_za_y1 = f.subs(x, minimum)
prvi_izvod_za_y2 = f.subs(x, maksimum)

minimum_tacka = [minimum, prvi_izvod_za_y1]
maksimum_tacka = [maksimum, prvi_izvod_za_y2]

print(minimum_tacka)
```

```
33.     print(maksimum_tacka)
```

Sympy je Python biblioteka za simboličku matematiku. Cilj mu je da bude alternativa sistemima kao što su Mathematica ili Maple, a da kod bude što jednostvniji. Sympy je u potpunosti napisan u Python-u i ne zahtijeva nikakve spoljne biblioteke.

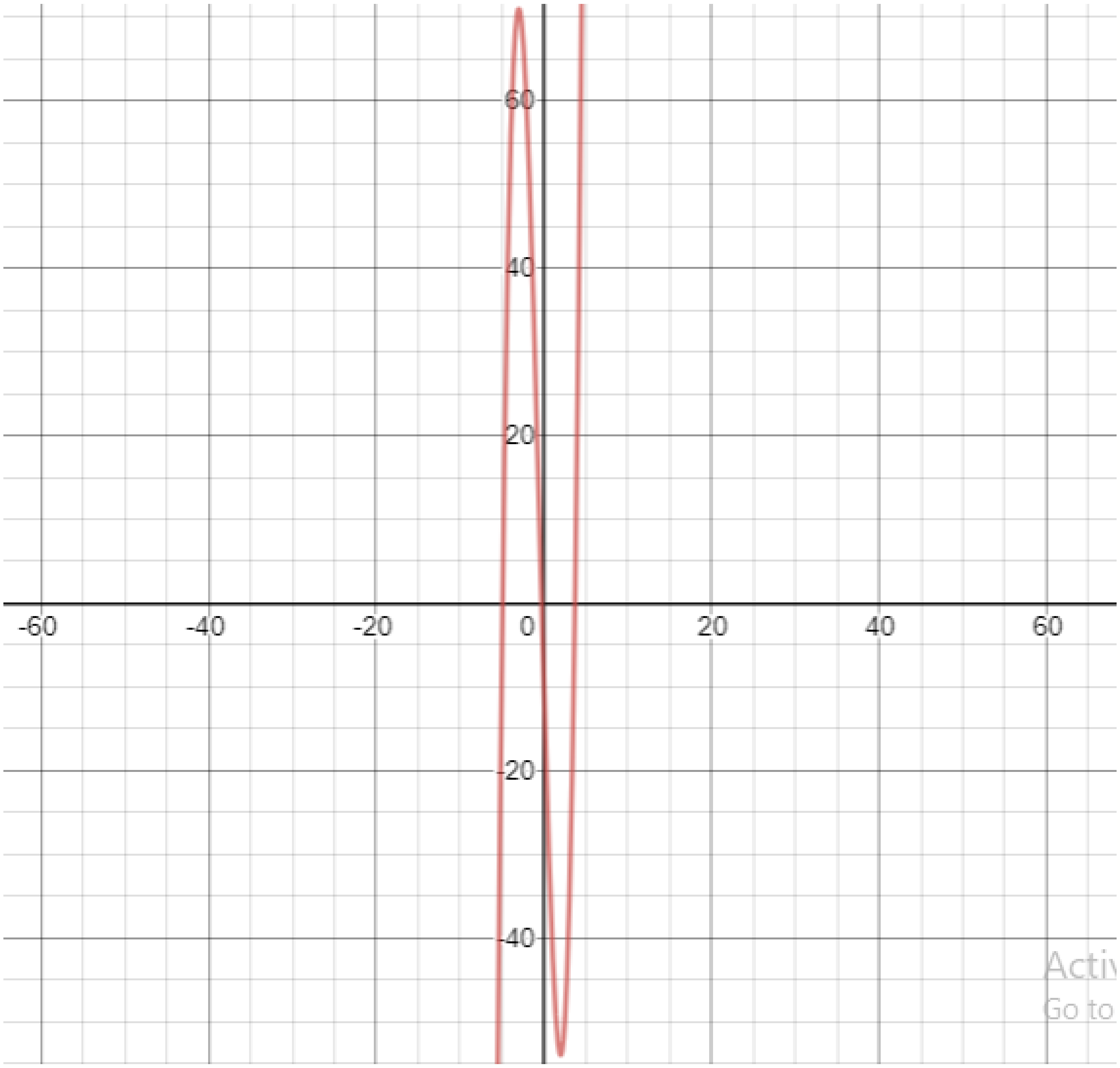
U redu broj 1 smo uveli iz biblioteke sympy * za množenje. Zatim smo x predstavili kao simbol koji predstavlja varijablu u funkciji. Za funkciju smo izabrali prethodni primjer.

Pomoću metode .diff(x) nad funkcijom f smo derivirali f, odnosno odredili njen prvi izvod i dobijamo $x^2+x-6=0$. Zatim u redu broj 5 je riješena jednačina $f'(x)=0$ iz koje smo dobili stacionarne tačke a to su [-3, 2], riješenja smo prikazali u obliku liste l, a te tačke kao x1 = -3 i x2 = 2.

Dalje, nalazimo drugi izvod pozivajući dva puta istu metodu .diff(x), time dobijamo 12x+6. U 9.-om i 10.-om redu se poziva metoda .subs() sa dva parametra od kojih je prvi naša varijabla x a drugi broj x1, to znači da dodajemo varijabli x broj x1, sada u drugoj derivaciji zamjenjujemo x sa x1 i time se riješava 12*(-3)+6, dobijamo broj -30.
Ponavljamo postupak i za broj x2 = 2 i dobijamo 30. U narednim linijama koda provjeravamo da li su dobijeni brojevi veći ili manji od nule i time nalazimo maksimalni x i minimalni x.

U 26.-toj liniji koda se tačke x1 i x2 vraćaju u početnu funkciju $f(x)$ što dobijamo minimum za y i maksimum za y. Za ispis tačke imamo sve što nam je potrebno, te je pozivom funkcije print() ispisujemo. U našem slučaju samo nas zanima maksimalna tačka i ona iznosi MAX(-3, 71).

Grafički prikaz funckije, vidimo da je naš maksimum tačan.



Amina Dacić
Prirodno-matematički fakultet
Teorijska kompjuterska nauka
5574/M

KRAJ