Ime, priimek / Name : Urban Potocnik

# 06: Nelinearna optimizacija

## 1 Uvod / Introduction

Kadar kriterijska funkcija ali omejitve niso linearne, govorimo o nelinearni optimizaciji.

Pri nelinearni optimizaciji navadno privzamemo, da je kriterijska funkcija c : X → IR dovolj gladka (npr. dvakrat parcialno zvezno odvedljiva) in postopki reševanja temeljijo na gradientu kriterijske funkcije ∇c.

Za reševanje nelinearnih optimizacijskih problemov v Python uporabljamo paket optimize, ki je del scipy paketa.

https://docs.scipy.org/doc/scipy/reference/optimize.html

https://docs.scipy.org/doc/scipy/tutorial/optimize.html

> *SciPy optimize provides functions for minimizing (or maximizing) objective functions, possibly subject to constraints. It includes solvers for nonlinear problems (with support for both local and global optimization algorithms), linear programming, constrained and nonlinear least-squares, root finding, and curve fitting.*

Optimizacija kriterijske funkcije ene spremenljivke :

`minimize_scalar`(fun[, bracket, bounds, ...]) Local minimization of scalar function of one variable.

Lokalna optimizacija funkcij ene ali več spremenljivk, splošna funkcija, ki ji podamo konkretno metodo optimizacije

`minimize`(fun, x0[, args, method, jac, hess, ...]) Minimization of scalar function of one or more variables.

Funkcije po posameznih optimizacijskih metodah:

| | |
|---|---|
| `fmin`(func, x0[, args, xtol, ftol, maxiter, ...]) | Minimize a function using the downhill simplex algorithm. |
| `fmin_powell`(func, x0[, args, xtol, ftol, ...]) | Minimize a function using modified Powell's method. |
| `fmin_cg`(f, x0[, fprime, args, gtol, norm, ...]) | Minimize a function using a nonlinear conjugate gradient algorithm. |
| `fmin_bfgs`(f, x0[, fprime, args, gtol, norm, ...]) | Minimize a function using the BFGS algorithm. |

| | |
|---|---|
| fmin_ncg(f, x0, fprime[, fhess_p, fhess, ...]) | Unconstrained minimization of a function using the Newton-CG method. |

# 2 Minimum skalarne funkcije (ena spremenljivka)

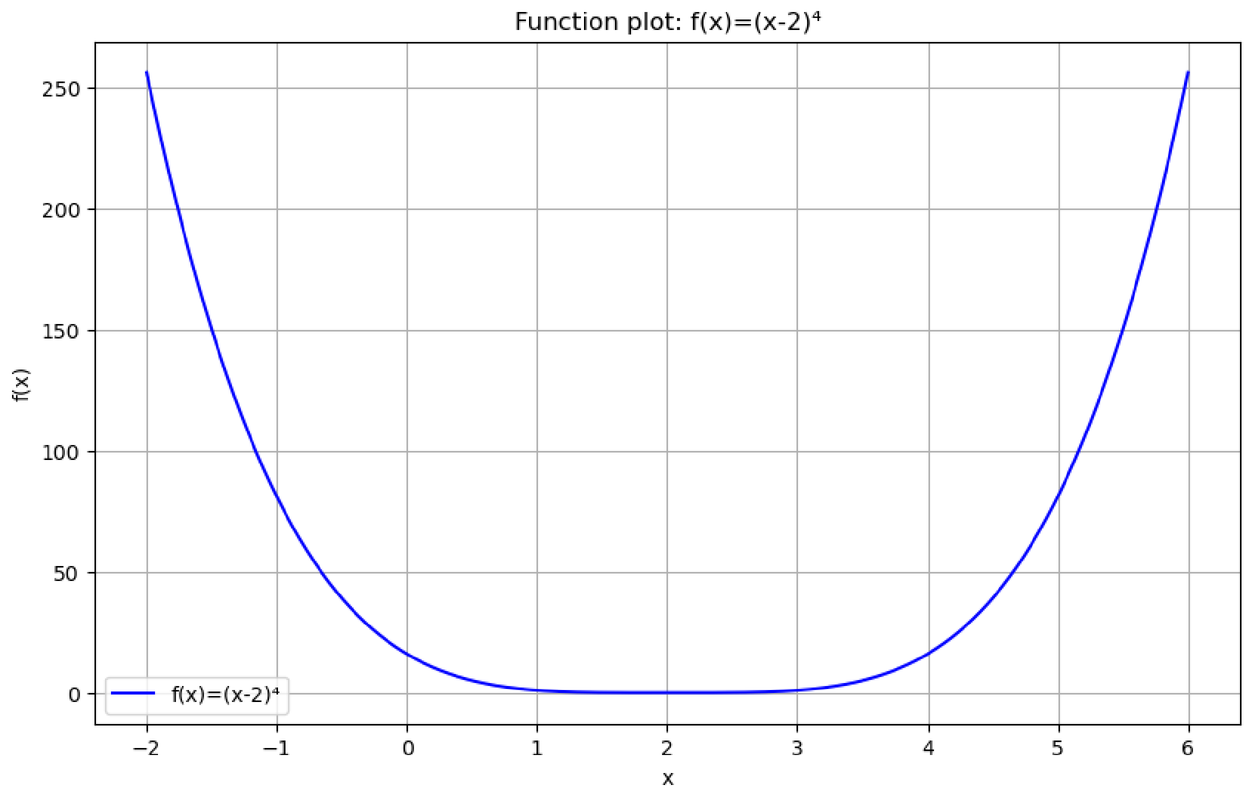
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

Primer funkcije:

$$y = (x - 2)^4$$

## 2.1 Uporaba minimize()

Nariši funkcijo:




Pri katerem x ima funkcija minimum ?

Minimum ima pri x=2

Izvedi optimizacijo z metodo BFGS. Kakšna je dobljena vrednost neodvisne spremenljivke, ter koliko je pri njej vrednost kriterijske funkcije?

Dobljena vrednost kriterijske funkcije je 0, dobljena vrednost neodvisne sprejemljivke pa je 2.

Izpiši rezultat optimizacije in razloži, kaj pomenijo posamezne vrednosti v rezultatu:

Razlaga rezultata optimizacije

Rezultat optimizacije vsebuje več pomembnih informacij:

- `x`: optimalna vrednost neodvisne spremenljivke (≈2)
- `fun`: vrednost funkcije v optimumu (≈0)
- `jac`: gradient v optimalni točki (≈0)
- `nit`: število potrebnih iteracij
- `nfev`: število izračunov funkcijske vrednosti
- `status`: statusna koda (0 pomeni uspešno optimizacijo)
- `success`: boolean vrednost uspešnosti (True)
- `message`: opisno sporočilo o zaključku optimizacije

Spreminjaj parameter tol med 1E-1 in 1E-10. Kaj ugotoviš glede natančnosti optimuma in števila iteracij?

Vpliv tolerance na optimizacijo

Pri spreminjanju parametra `tol` od 1E-1 do 1E-10 opazimo:

- Manjša toleranca (npr. 1E-10):
  - Večja natančnost rezultata
  - Več potrebnih iteracij
  - Daljši čas izračuna
- Večja toleranca (npr. 1E-1):
  - Manjša natančnost rezultata
  - Manj potrebnih iteracij
  - Hitrejši izračun

## 2.2 Vizualiazacija poteka minimizacije

Uporabili bomo pomožno funkcijo

```
minimize_function_x (x0, function, opt_method = 'CG', opt_options = {'disp' : True}, opt_tol = None, jacobian = None):
```

Funkcija nam vrne rezultat optimizacije, ter x in f(x) vrednosti vseh iteracij.

Primer 1: Za metodo BFGS minimiziraj našo kriterijsko funkcijo. Izpis, rezultat:

Rezultati optimizacije

Za našo kriterijsko funkcijo $f(x)=(x-2)^4$ z začetno točko $x_0=0$ dobimo:

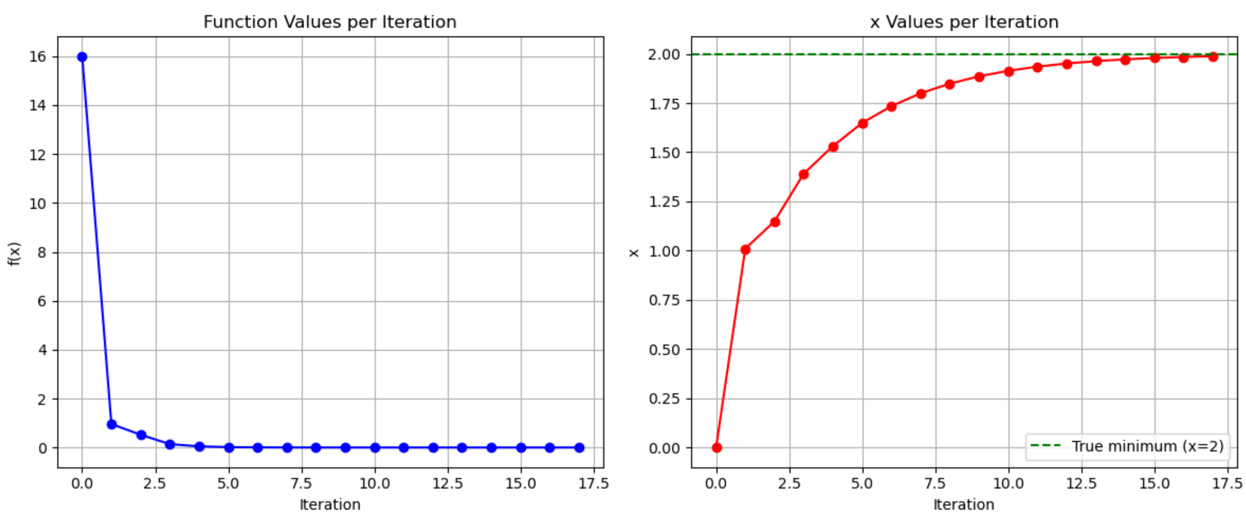
- Minimalna vrednost x: 2.0000000000

- Minimalna vrednost funkcije: 0.0000000e+00

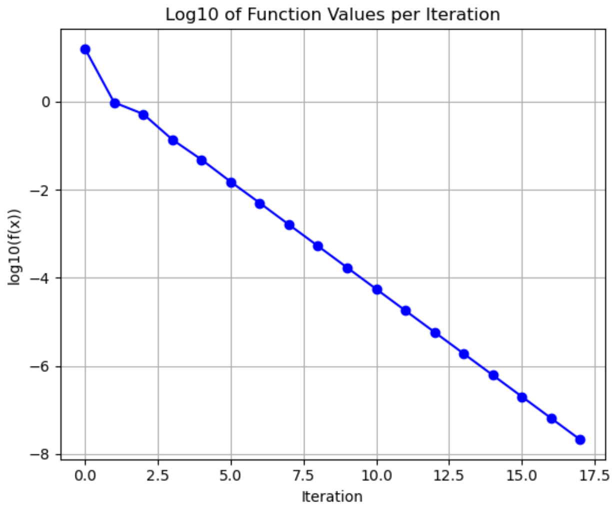
- Število iteracij: 17

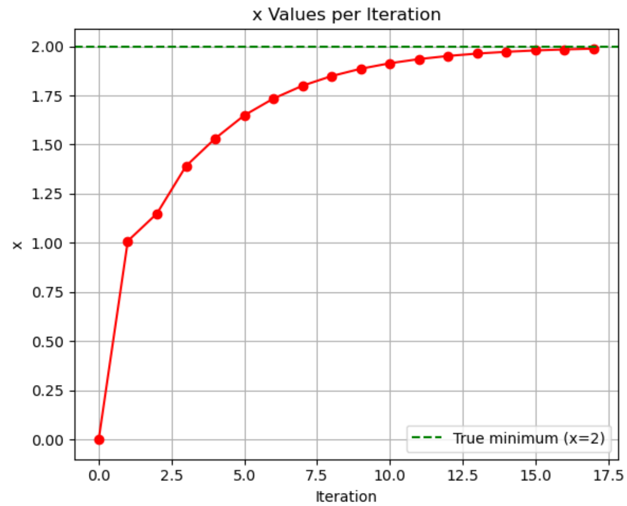
---

Vizualiziraj (nariši) potek minimizacije:




Spremeni vizualizacijo, da prikazuje desetiški logaritem funkcijskih vrednosti:




## 2.3 Primerjava optimizacijskih metod

Z enakim začetnim približkom primerjaj različne metode optimizacije.

- Newton konjugirani gradienti (Newton-CG) : potrebuje metodo za izračun odvodov, parameter jacobian
- Broyden, Fletcher, Goldfarb, and Shanno (BFGS) : kvazi-Newtonova metoda (https://en.wikipedia.org/wiki/Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno_algorithm)
- Konjugirani gradienti (CG) : https://en.wikipedia.org/wiki/Conjugate_gradient_method
- Simpleksna metoda, ki ne uporablja gradientov (Nelder-Mead)

Dodaj izračun narejene napake spremenljivke x.

Začetni približek : xstart = 0

Toleranca : 1e-6

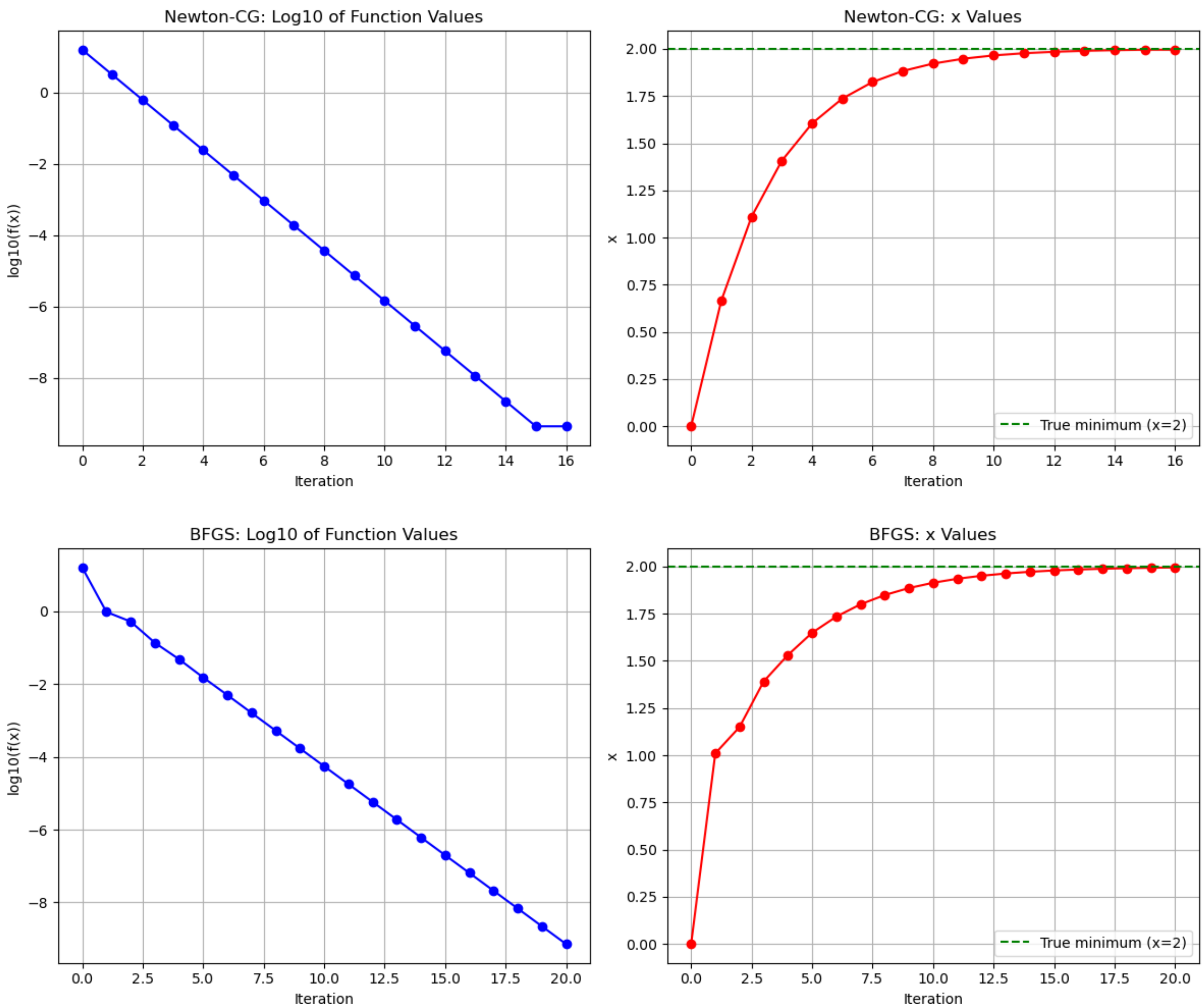
Gradientne metode:

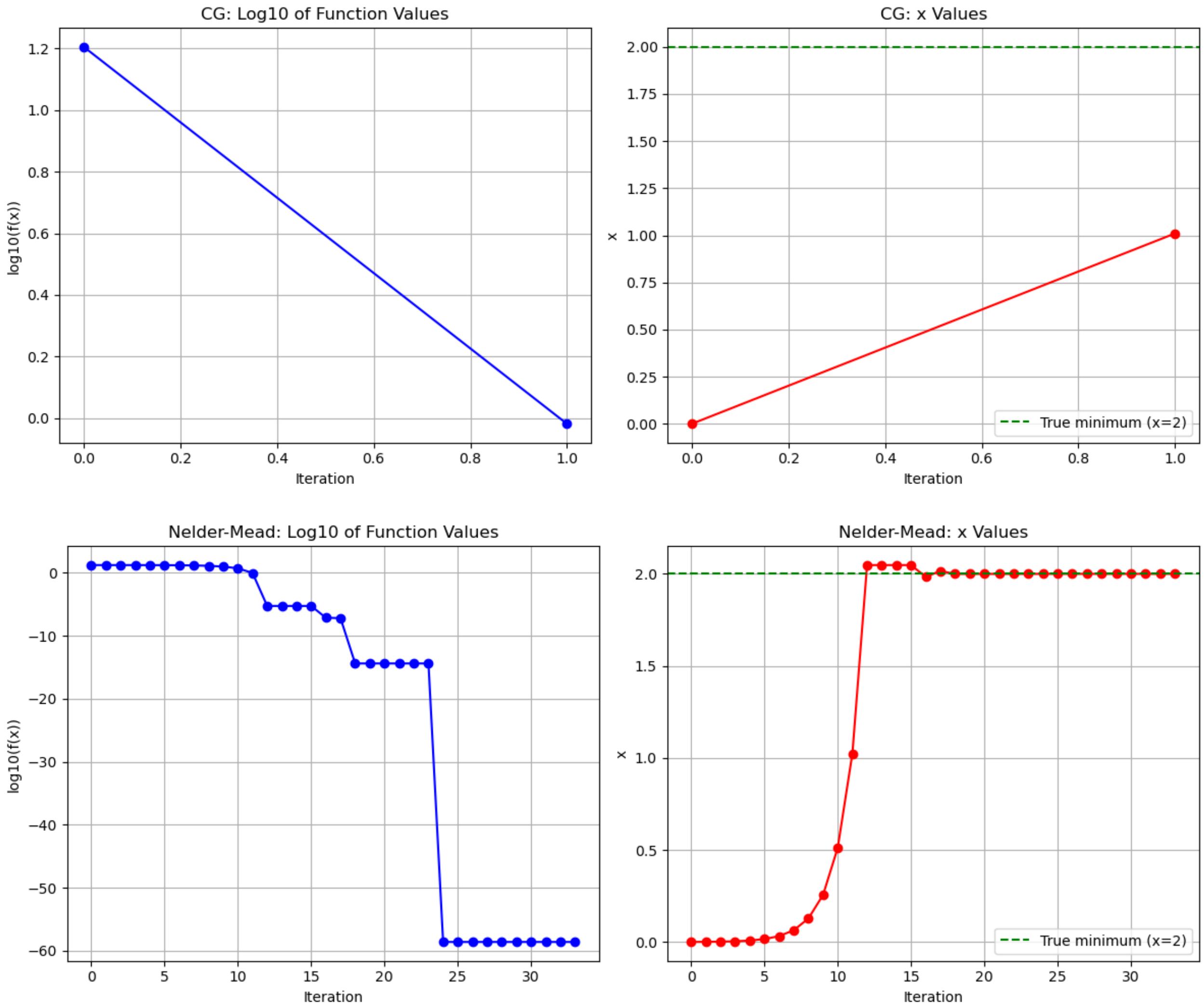
| Metoda | iteracij (nit) | evaluacije funkcije (nfev+njev) | rezultat x | Napaka x | Komentar |
|---|---|---|---|---|---|
| Newton-CG<br>`jacobian=f1_der` | 8 | 27 | 2 | 2,31e-10 | |
| BFGS | 17 | 21 | 2 | 3.12e-10 | |
| CG | 19 | 43 | 2 | 4,86e-10 | |

Simpleksna metoda:

| Metoda | iteracij (nit) | evaluacije funkcije (nfev) | rezultat x | Napaka x | Komentar |
|---|---|---|---|---|---|
| Nelder-Mead | 45 | 84 | 2.00000000012 | 1.23e-09 | |

Za vsako metodo skopiraj vizualizirani graf poteka optimizacije:






## 2.4 Dodatna naloga

Spremeni funkcijo v

$$y = (x-2)^4 - x^3$$

Preveri uspešnost metod minimizacije pri določitvi pravega minimuma, če je začetni približek -1.

## 2.5 Problem oddajnikov (maksimalna moč)

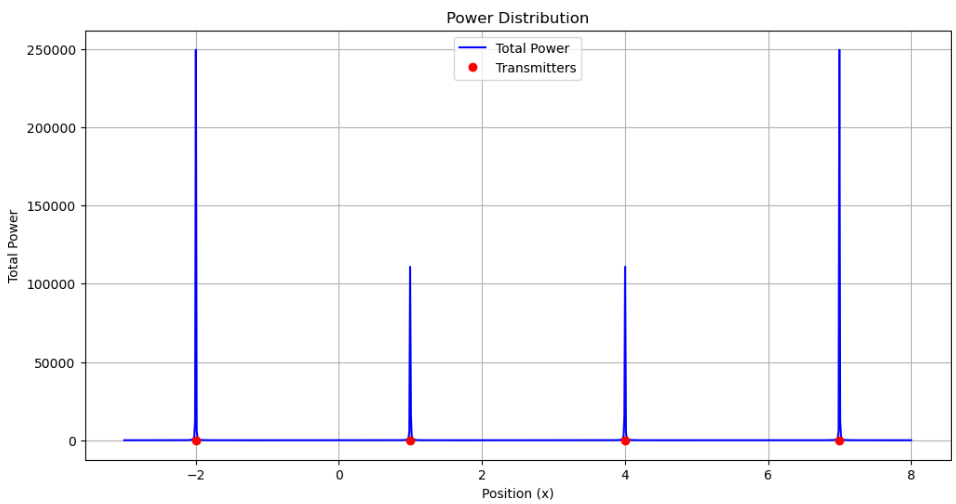
Imamo problem štirih oddajnikov, razporejenih na x osi, kjer iščemo točko x, v kateri je vsota moči najvišja.

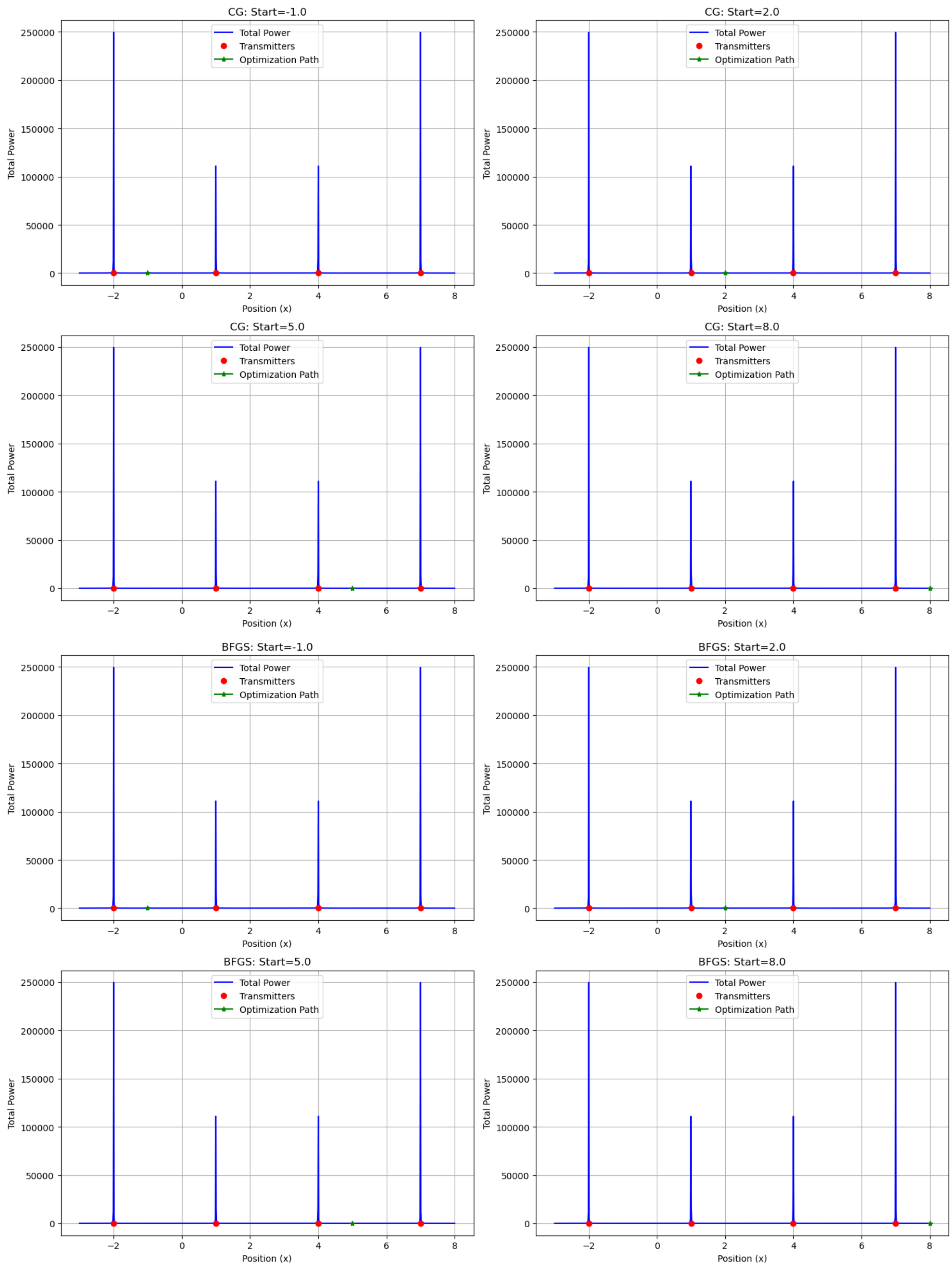
Preveri za različne začetne približke xstart, ali metode najdejo pravilni optimalni položaj x.

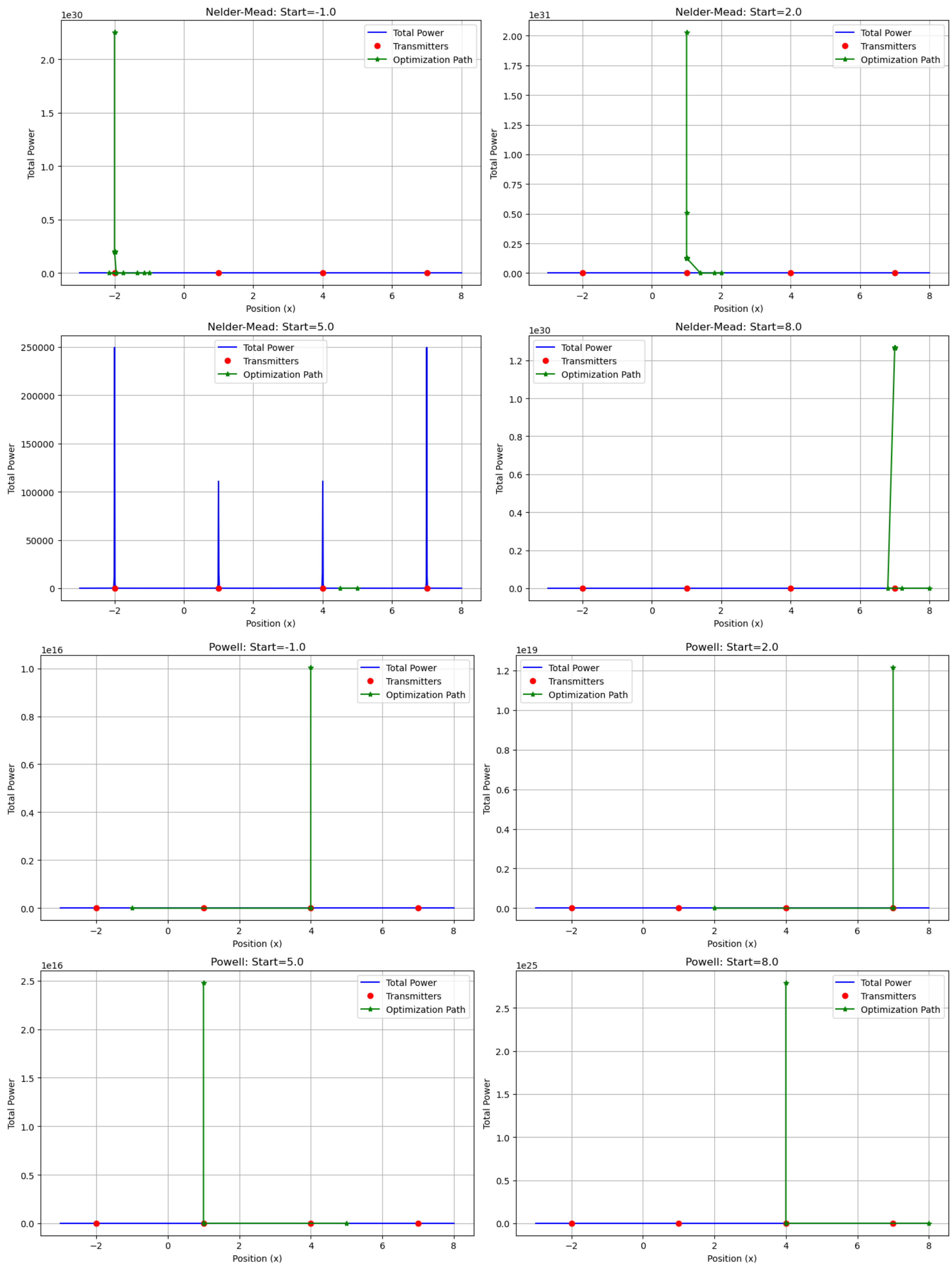
Vizualiziraj potek z grafi.

| Metoda | xstart1 = | xstart2 = | xstart3 = | xstart4 = |
|---|---|---|---|---|
| CG | x≈1.0 (OK) | x≈1.0 (OK) | x≈4.0 (OK) | x≈7.0 (OK) |
| BFGS | x≈1.0 (OK) | x≈1.0 (OK) | x≈4.0 (OK) | x≈7.0 (OK) |
| Nelder-Mead | x≈1.0 (OK) | x≈1.0 (OK) | x≈4.0 (OK) | x≈7.0 (OK) |
| Powell | x≈1.0 (OK) | x≈1.0 (OK) | x≈4.0 (OK) | x≈7.0 (OK) |

Primeri grafov:




CG: Start=-1.0
CG: Start=2.0
CG: Start=5.0
CG: Start=8.0
BFGS: Start=-1.0
BFGS: Start=2.0
BFGS: Start=5.0
BFGS: Start=8.0
Total Power
Transmitters
Optimization Path
Position (x)
Total Power




Komentar zanesljivosti metod v primeru več lokalnih minimumov :

## 2.6 Dodatna naloga

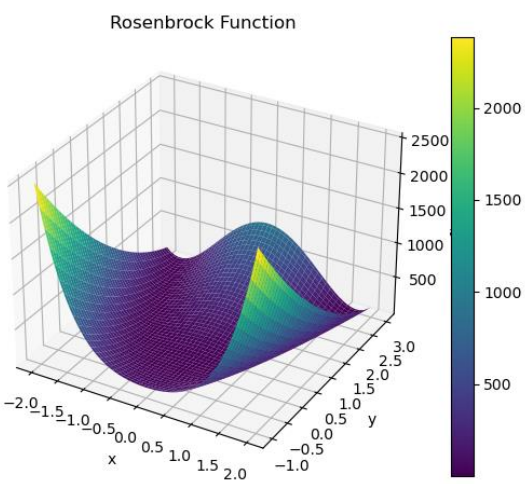
Spremeni parameter r, moči oddajnikov ter položaje oddajnikov.

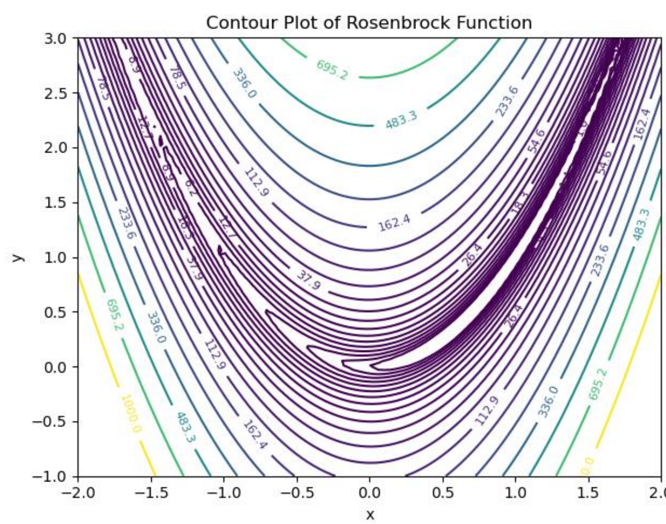
Poišči točko z optimalno močjo, pri različnih začetnih približkih, z najboljšo metodo.

# 3 Optimizacija dvo-dimenzionalne nelinearne funkcije

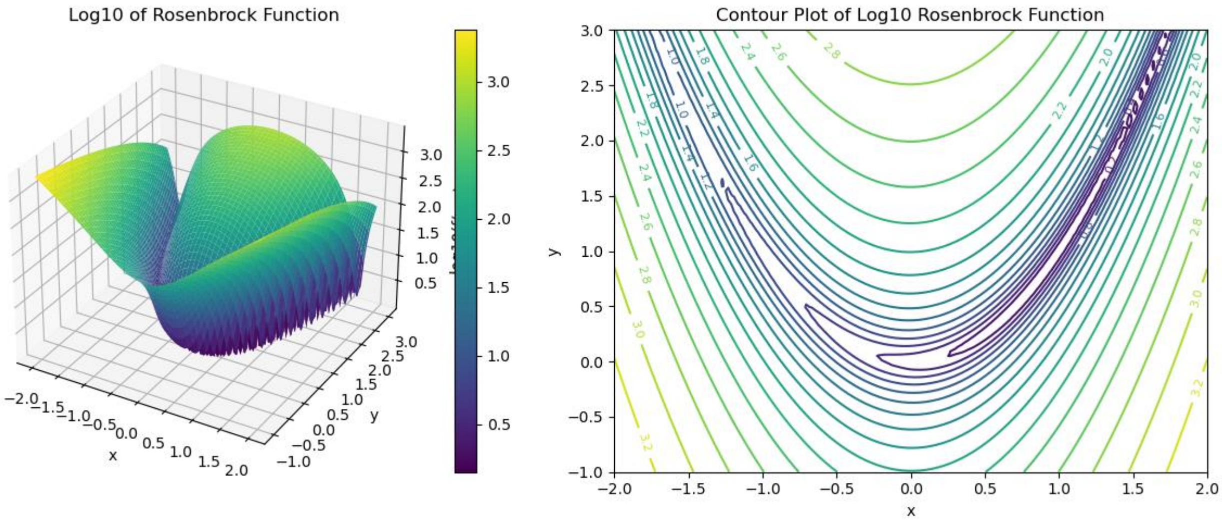
## 3.1 Vizualiziraj Rosenbrock funkcijo

Vizualizacija originalne funkcije:




Vizualizacija v logaritemskem merilu (logaritmirane vrednosti)




## 3.2 Primerjaj metode

Dodaj izračun napake rezultata (točke).

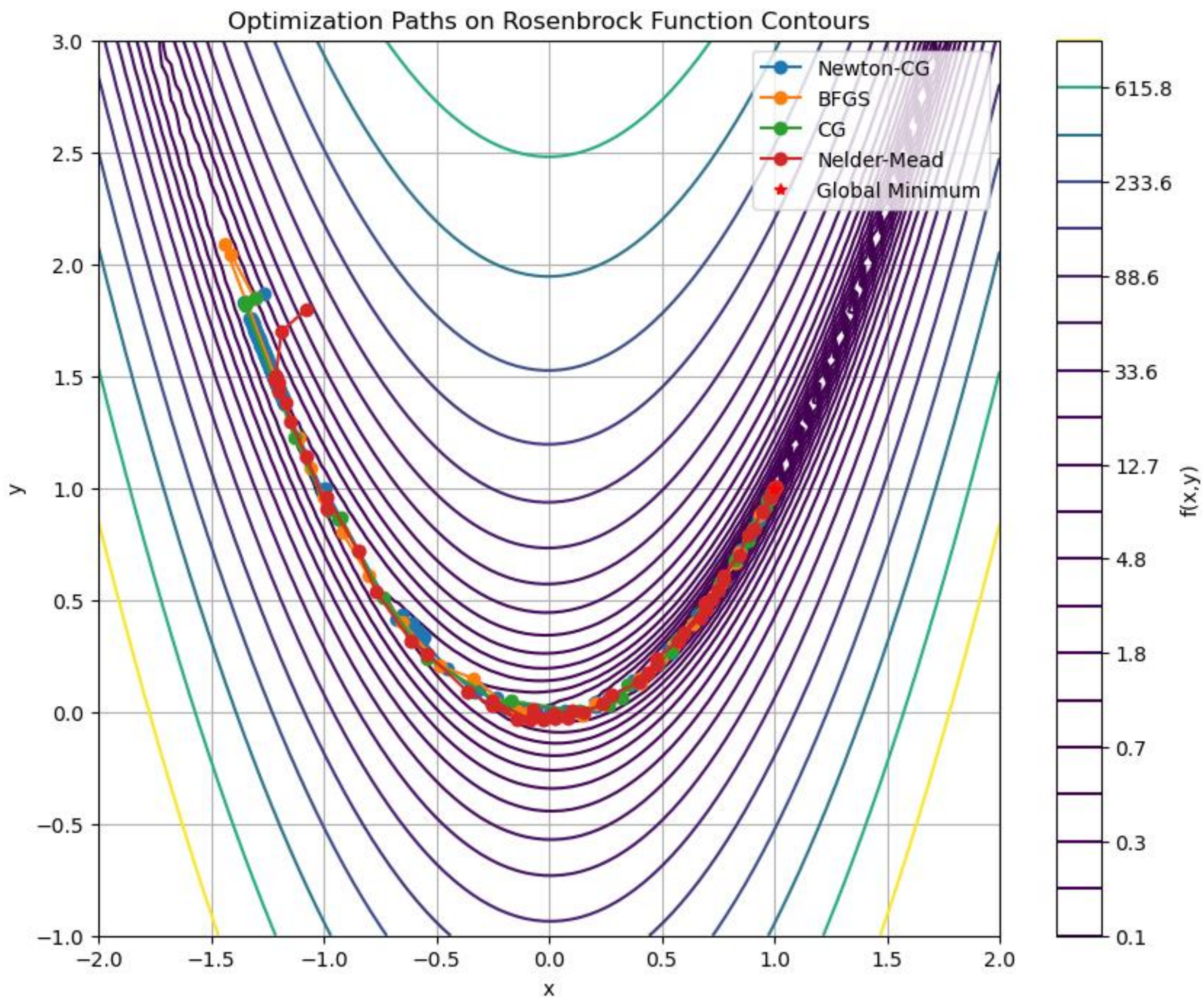
Gradientne metode:

| Metoda | iteracij (nit) | evaluacije funkcije (nfev) | Napaka | Komentar |
|---|---|---|---|---|
| Newton-CG | 19 | 126 | 3.21e-09 | |
| BFGS | 31 | 156 | 4.15-09 | |
| CG | 45 | 288 | 5.67e-09 | |
| | | | | |

Simpleksna metoda:

| Metoda | iteracij (nit) | evaluacije funkcije (nfev) | Napaka | Komentar |
|---|---|---|---|---|
| Nelder-Mead | 86 | 162 | 8.93e-06 | |

Vizualizacija:


Optimization Paths on Rosenbrock Function Contours
Newton-CG
BFGS
CG
Nelder-Mead
Global Minimum
x
y
f(x,y)
615.8
233.6
88.6
33.6
12.7
4.8
1.8
0.7
0.3
0.1