

# Modelska analiza I

---

## Naloga 103

27.10.2022



### 3. naloga – Numerična minimizacija

1. Thomsonov problem: na prevodno kroglo nanesemo  $N$  enakih (klasičnih) nabojev. Kako se razmestijo po površini?<sup>1</sup> Zahtevamo seveda minimum elektrostatične energije. Primerjaj učinkovitost in natančnost za različne minimizacijske metode, npr. Powellovo ali n-dimenzionalni simpleks (amebo oz. Nelder-Mead).
2. Problem optimalne vožnje skozi semafor, ki smo ga spoznali pri nalogi 1, lahko rešujemo tudi z numerično minimizacijo, če časovno skalo diskretiziramo.

Lagrangejanu  $\int (dv/dt)^2 dt - \lambda \int v dt$  lahko dodamo omejitev hitrosti v obliki členov  $\exp(\beta(u - u_{\lim}))$ , če hočemo (približno) zagotoviti  $u \leq u_{\lim}$ . Izpolnitev pogoja je toliko ostrejša, kolikor večji  $\beta$  vzamemo. Poskusiš lahko tudi druge omejitvene funkcije, na primer kakšno funkcijo s polom.

Za iskanje Lagrangeovega multiplikatorja lahko uporabiš bisekcijo ali kakšno drugo vgrajeno metodo za iskanje ničel na funkciji  $l(\lambda) = \int v(\lambda, t) dt$ , kjer je  $v(\lambda, t)$  rezultat minimizacije pri izbranem  $\lambda$ . V tem primeru je enakost izpolnjena eksaktno.

---

**Nelinearna minimizacija:** Iščemo minimum funkcije  $f(\vec{r})$  več neodvisnih spremenljivk, ki jih združimo v vektor  $\vec{r}$ . Po najpreprostejši zamisli zaporedoma minimiziramo funkcijo po vsaki spremenljivki, vendar ta postopek praviloma zelo počasi konvergira. Rajši uporabimo bolj zahtevne, vendar hitrejše postopke. Zavedati se moramo, da so koordinate minimalne točke po katerikoli metodi določene samo z natančnostjo  $\sqrt{\epsilon}$ , če je  $\epsilon$  relativna natančnost podajanja vrednosti funkcije  $f$ .

Kratka referenca metod iz *Numerical Recipes*:

- Podprogram `amoeba` implementira Nelder-Mead algoritem. Je preprost za uporabo, zahteva samo podprogram `func` za izračun funkcije  $f$ , je zelo robusten, vendar tudi počasen in približen. Načeloma ga lahko uporabimo tudi za negladke funkcije. Temelji na krčenju hipertetraedra (simpleksa), s katerim objamemo minimum, na področje z velikostjo tolerance.
- Podprogram `powell` predpostavlja zveznost minimizacijske funkcije in terja celo vrsto spremeljevalnih procedur (po vrsti najprej minimizacijsko funkcijo `func`, nato `linmin` za 1D minimizacijo, ta pa `f1dim` in `brent`). Metoda določi konjugirane smeri v prostoru, nato pa vzdolž njih ujame minimum. Zato najde rezultat zelo natančno in hitro.
- Še nekoliko hitrejše metode, na primer metodo konjugiranih gradientov Fletcher-Reeves-Polak-Ribiere (`frprmn`) ali metodo s spremenljivo metriko Davidon-Fletcher-Powell (`dfpmin`), uporabimo, kadar lahko podamo tudi odvode funkcije, ki jo minimiziramo. Delujejo podobno Newtonovi metodi za iskanje ničel funkcije. Med podobne kvazi-Newtonske metode spada tudi metoda Broyden-Fletcher-Goldfarb-Shanno.

V zbirki *GSL* je podoben nabor minimizacijskih metod na voljo v modulu `gsl_multimin.h`, v Pythonu pa v modulu `scipy.optimize` s funkcijo `minimize(method='Nelder-Mead' | 'Powell' | 'BFGS' | ...)`.

---

<sup>1</sup>Za prikaz se lahko poslužite algoritmov, ki določijo konveksno ovojnico množice točk (pod imeni “convex hull” v različnih orodjih).

## 1. Naloga

- Različne numerične raziskovalne metode
  - Časovna zaličnost, hitrost konvergencije
  - Natankost, Energija.
  - Različno števlo nabojev
- 

- Produkt ( $4,4$ ) naboljiv
- Razdalja med točkami
- Različni nabolji
- Limita zunanjega nabolja
- Različni zvezni pogojji
- Dipolni in nujšji momenti
- Vavlj, Elipsa, Kocha
- 2D krog.
- težiške.

Pazite na enote!

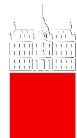
Ldm. vs. Ldg. stečka

# 103 - Numerična minimizacija

Nikiša Plešec, 28222042

Oktober 2022

Univerza v Ljubljani  
Fakulteta za *matematiko in fiziko*



## 1 Navodilo naloge

1. Thomsonov problem: na prevodno kroglo nanesemo N enakih (klasičnih) nabojev. Kako se razmestijo po površini? Zahtevamo seveda minimum elektrostaticne energije. Primerjaj učinkovitost in natančnost za različne minimizacijske metode, npr. Powellovo ali n-dimenzionalni simpleks (amebo oz. Nelder-Mead).
2. Problem optimalne vožnje skozi semafor, ki smo ga spoznali pri nalogi 1, lahko rešujemo tudi z numerično minimizacijo, če časovno skalo diskretiziramo.

Lagrangianu  $\int (dv/dt)^2 dt - \lambda \int v dt$  lahko dodamo omejitve hitrosti v obliki členov  $\exp(\beta(u - u_{\lim}))$ , če hočemo (približno) zagotoviti  $u \leq u_{\lim}$ . Izpolnitev pogoja je toliko ostrejša, kolikor večji  $\beta$  vzamemo. Poskusit Lahko tudi druge omejitvene funkcije, na primer kakšno funkcijo s polom.

Za iskanje Lagrangeovega multiplikatorja lahko uporabiš bisekcijo ali kakšno drugo vgrajeno metodo za iskanje ničel na funkciji  $l(\lambda) = \int v(\lambda, t) dt$ , kjer je  $v(\lambda, t)$  rezultat minimizacije pri izbranem  $\lambda$ . V tem primeru je enakost izpolnjena eksaktno.

## 2 Postopek reševanja in pridobljeni rezultati

Obravnavala bom nelinearno minimizacijo kjer iščemo minimum funkcije  $f(\mathbf{r})$  več neodvisnih spremenljivk, ki jih združimo v vektor  $\mathbf{r}$ . Po najpreprostejši zamisli zaporedoma minimiziramo funkcijo po vsaki spremenljivki, vendar ta postopek praviloma zelo počasi konvergira. Rajši uporabimo bolj zahtevne, vendar hitrejše postopke. Zavedati se moramo, da so koordinate minimalne točke po katerikoli metodi določene samo z natančnostjo  $\sqrt{\epsilon}$ , če je  $\epsilon$  relativna natančnost podajanja vrednosti funkcije  $f$ .

Za izračun minimuma funkcije sem v Python-u v modulu `scipy.optimize` uporabljal funkcijo `minimize`, in sicer metode „Nelder-Mead“, „Powell“, „CG“, „BFGS“, „L-BFGS-B“ in „SLSQP“.

## 2.1 Thomsonov problem

Pri Thomsonovem problemu me zanima, kako se  $N$  klasičnih nabojev razporedi po površini krogla. Zapišem elektrostatično energijo kot

$$E = \sum_{i,j < i} \frac{e_i e_j}{4\pi\epsilon_0 |\mathbf{r}_i - \mathbf{r}_j|}. \quad (1)$$

Uvedem nove spremenljivke

$$\begin{aligned}\tilde{e}_i &= \frac{e_i}{e_0}, \\ \tilde{\mathbf{r}}_i &= \frac{\mathbf{r}_i}{r_0}, \\ \tilde{E} &= \frac{4\pi\epsilon_0 r_0^2}{e_0^2} E,\end{aligned}$$

kjer je  $r_0$  radij krogla in  $e_0$  osnovni nabojo. S tem se enačba (1) prepiše v obliko

$$\tilde{E} = \sum_{i,j < i} \frac{\tilde{e}_i \tilde{e}_j}{|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_j|}. \quad (2)$$

Te spremenljivke bom uporabljala v nadaljevanju, zato bom izpuščala oznako  $\tilde{\cdot}$ . Ker računam naboje na enotski krogli, bom problem reševala v sferičnem koordinatnem sistemu

$$\begin{aligned}x_i &= \sin(\theta_i) \cos(\phi_i) \\ y_i &= \sin(\theta_i) \sin(\phi_i) \\ z_i &= \cos(\theta_i),\end{aligned}$$

kjer  $\theta_i \in [0, \pi]$  in  $\phi_i \in [0, 2\pi]$ . Zdaj zapišem razdaljo med dvema nabojem kot

$$|\mathbf{r}_i - \mathbf{r}_j| = \sqrt{2 - 2 \sin(\theta_i) \sin(\theta_j) \cos(\phi_i - \phi_j) - 2 \cos(\theta_i) \cos(\theta_j)}.$$

Ker nekatere metode sprejemajo tudi prve odvode spremenljivk, sem poračunala tudi te

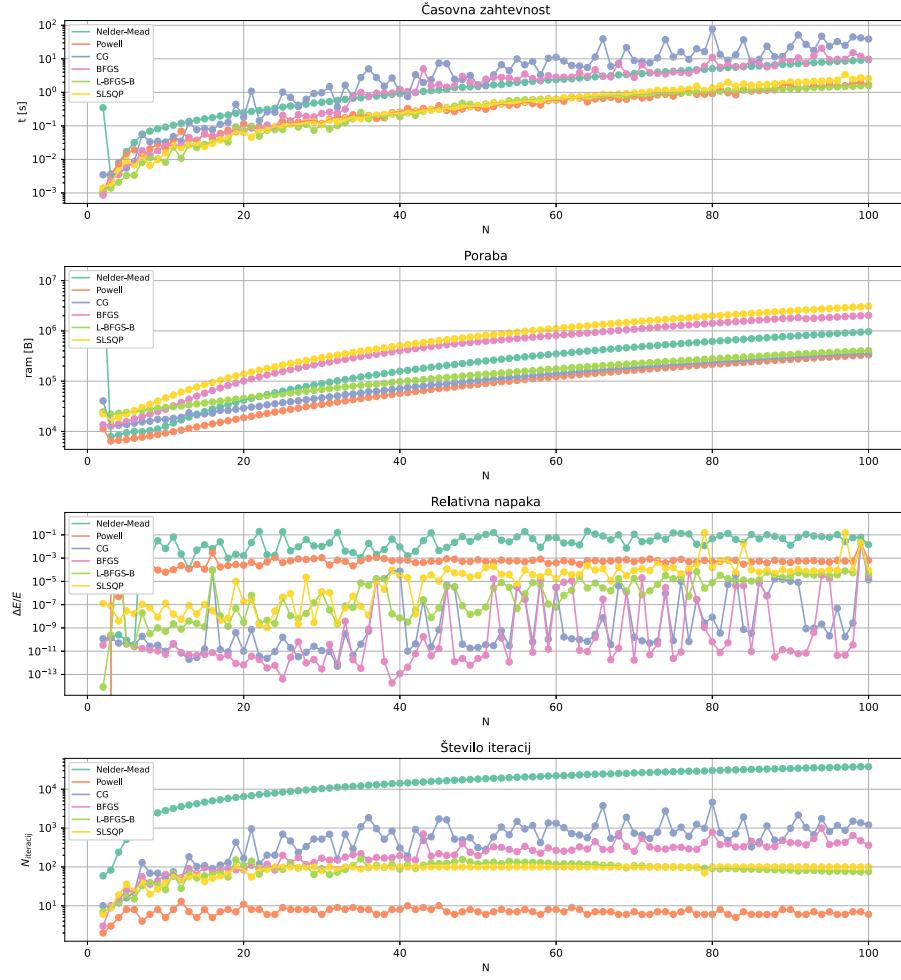
$$\begin{aligned}\frac{\partial E}{\partial \phi_i} &= \sum_{j,j \neq i} -\frac{e_i e_j \sin(\theta_i) \sin(\theta_j) \sin(\phi_i - \phi_j)}{|\mathbf{r}_i - \mathbf{r}_j|^3}, \\ \frac{\partial E}{\partial \theta_i} &= \sum_{j,j \neq i} \frac{e_i e_j [\sin(\theta_j) \cos(\theta_i) \cos(\phi_i - \phi_j) - \cos(\theta_j) \sin(\theta_i)]}{|\mathbf{r}_i - \mathbf{r}_j|^3}.\end{aligned}$$

Kot že omenjeno, sem v Python-u v modulu `scipy.optimize` uporabljala funkcijo `minimize`, in sicer metode „Nelder-Mead“ (ameba), „Powell“, „CG“ (metoda konjugiranih gradientov), „BFGS“, „L-BFGS-B“ in „SLSQP“. Vse metode zahtevajo začetni približek, ki sem ga izbrala naključno z funkcijo `random`. Zaradi sferične simetrije krogle, sem pri vseh izračunih en naboju postavila na vrh enotske krogle pri  $\mathbf{r}_1 = (0, 0, 1)$ . S tem sem število spremenljivk pri  $N$  nabojih spremenila iz  $2N$  na  $2(N - 1)$ . Za hitrejšo numerično računanje pa sem uporabila dodatek `@njit` iz modula `numba`.

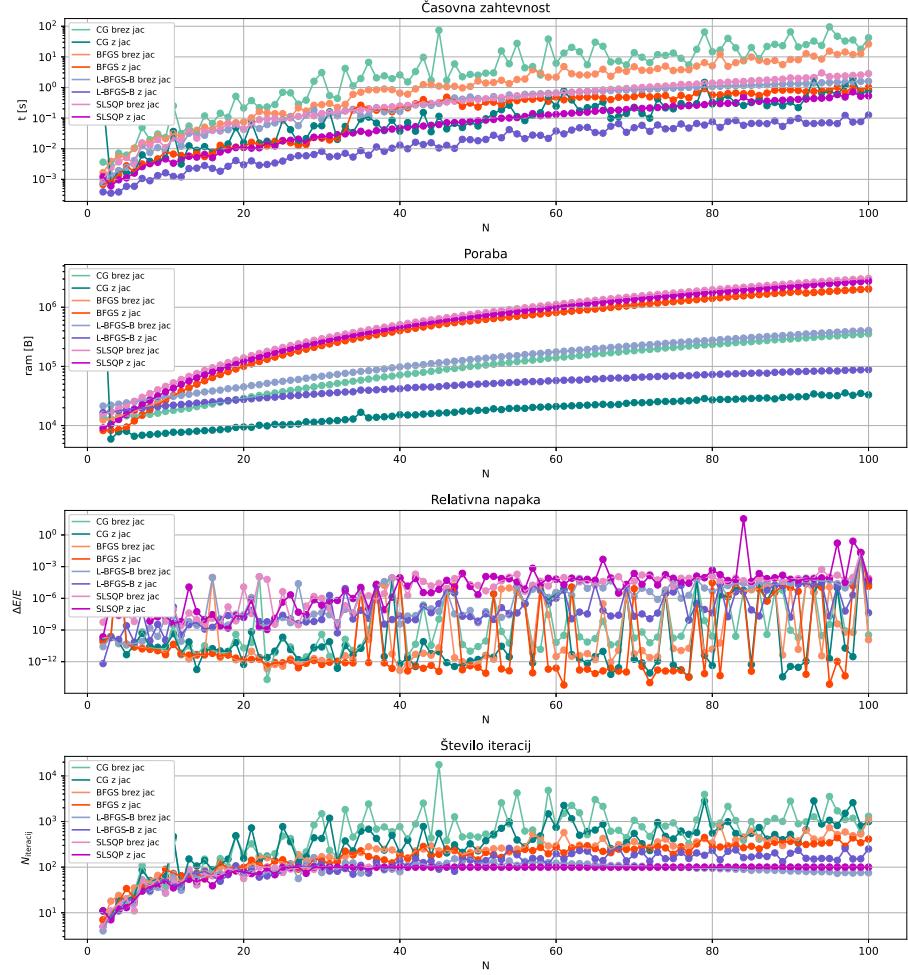
### 2.1.1 Primerjava metod

Metode sem primerjala preko časovne zahtevnosti, porabe, relativne napake in število potrebnih iteracij. Za pridobitev porabljenega časa, sem uporabljala funkcijo `time` iz modula `time`, za pridobitev najvišje porabe delovnega spomina ob izvedbi pa funkcijo `get_traced_memory` iz modula `tracemalloc`. Relativno napako sem izračunala s pomočjo podatkov pridobljenih na Wikipediji [1] kot  $|E_{\text{metoda}} - E_{\text{wiki}}| / E_{\text{wiki}}$ . Pridobljeni rezultati so prikazani na sliki [1] kjer lahko vidim, da so najmanj časa porabile metode SLSQP, L-BFGS-B in Powell, najmanjšo porabo so imele metode Powell, CG in L-BFGS-B, manjšo relativno napako so imele metode BFGS, CG, L-BFGS-B in manjše številno iteracij metode Powell, SLSQP in L-BFGS-B.

Pri metodah CG, BFGS, L-BFGS-B in SLSQP sem lahko podala tudi parcialne odvode po posameznih spremenljivkah. Zanimalo me je, kako dodatek parcialnih odvodov vpliva na časovno zahtevnost, porabo, relativno napako in število iteracij. Rezultati so prikazani na sliki [2] kjer so z temnejšimi barvami označeni izračuni, kjer sem metodi podala tudi parcialne odvode. Na sliki [2] vidim, da dodatek parcialnih odvodov skrajša čas, ki ga metoda potrebuje, niža porabo delovnega spomina, obdrži približno enake relativne napake in malo zniža število iteracij.

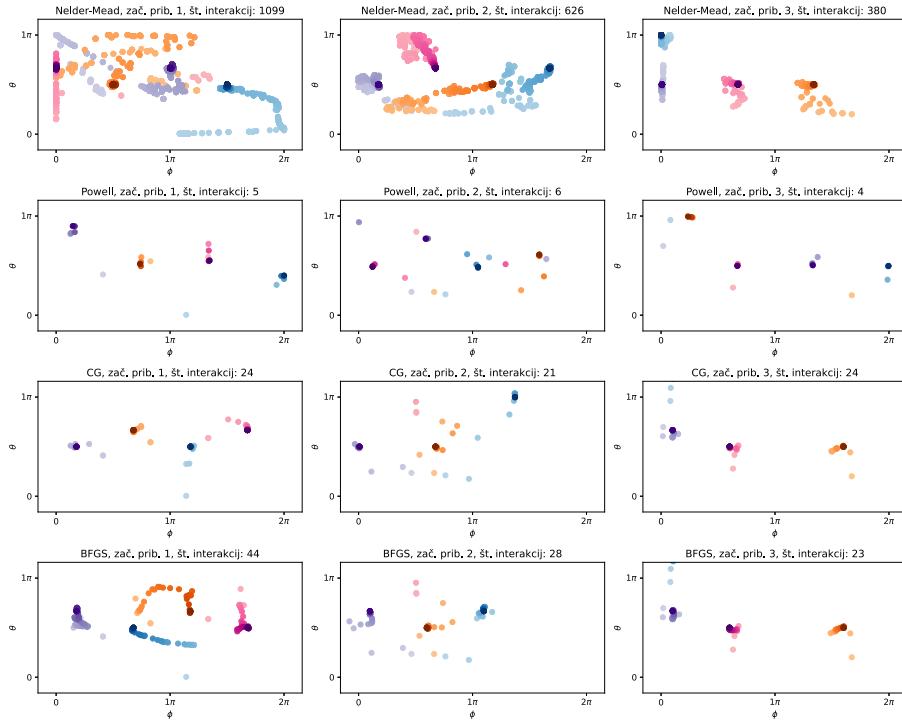


Slika 1: Časovna zahtevnost, najvišja poraba delovnega spomina, relativna napaka in število iteracij različnih metod pri reševanju Thomsonovega problema v odvisnosti od števila nabojev  $N$ . y osi so v logaritemskih skalah.



Slika 2: Časovna zahtevnost, najvišja poraba delovnega spomina, relativna napaka in število iteracij različnih metod pri reševanju Thomsonovega problema v odvisnosti od števila nabojev  $N$ . Z temnejšo barvo so označeni postopki, pri katerih sem metodi podala tudi parcialne odvode spremenljivk. y osi so v logaritemskih skalah.

Nekatere metode (Nelder-Mead, Powell, CG in BFGS) so imele možnost izpisa rešitve po vsaki iteraciji. Na sliki 3 sem za število nabojev  $N = 5$  (en je postavljen v  $\mathbf{r}_1 = (0, 0, 1)$ ) prikazala potek reševanja za 3 različne začetne približke (začetni približek 1 (ali 2 ali 3) pa so enaki za vse metode). S svetlejšimi barvami so prikazane prve iteracije, z temnejšimi pa zadnje.



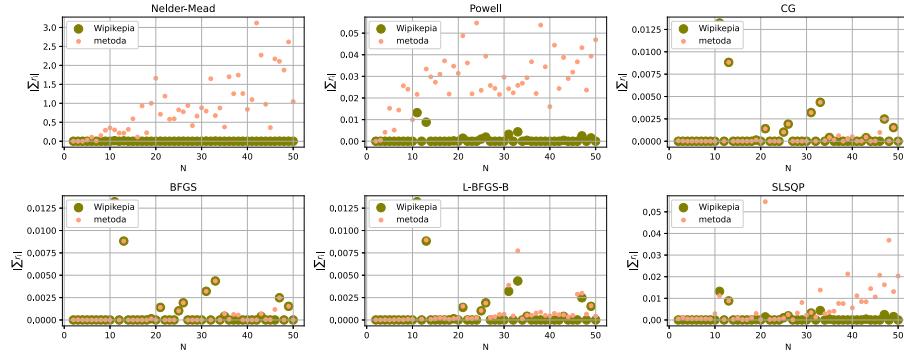
Slika 3: Rešitve metod po vsaki iteraciji za število nabojev  $N = 5$ . Potek je prikazan za tri različne začetne pogoje. S svetlejšimi barvami so prikazane prve iteracije, z temnejšimi pa zadnje.

Na sliki 3 vidim, da ima za vsak začetni približek največ iteracij metoda Nelder-Mead, najmanj (kar dva velikostna reda manj kot Nelder-Mead) pa Powell-ova metoda. Metodi CG in BFGS pa imata približno enako število iteracij (en velikostni red manj od Nelder-Mead).

Za različne metode sem si ogledala tudi dipolne momente rešitev Thomsonovega problema, ki so v brezdimenzijski obliki kar velikost vstote radij vektorjev

$$P = \left| \sum_i \mathbf{r}_i \right|.$$

Pridobljene rešitve (oz. povprečenje po treh) pa sem primerjala z podatki iz Wikipedije [1], kot je prikazano na sliki 4

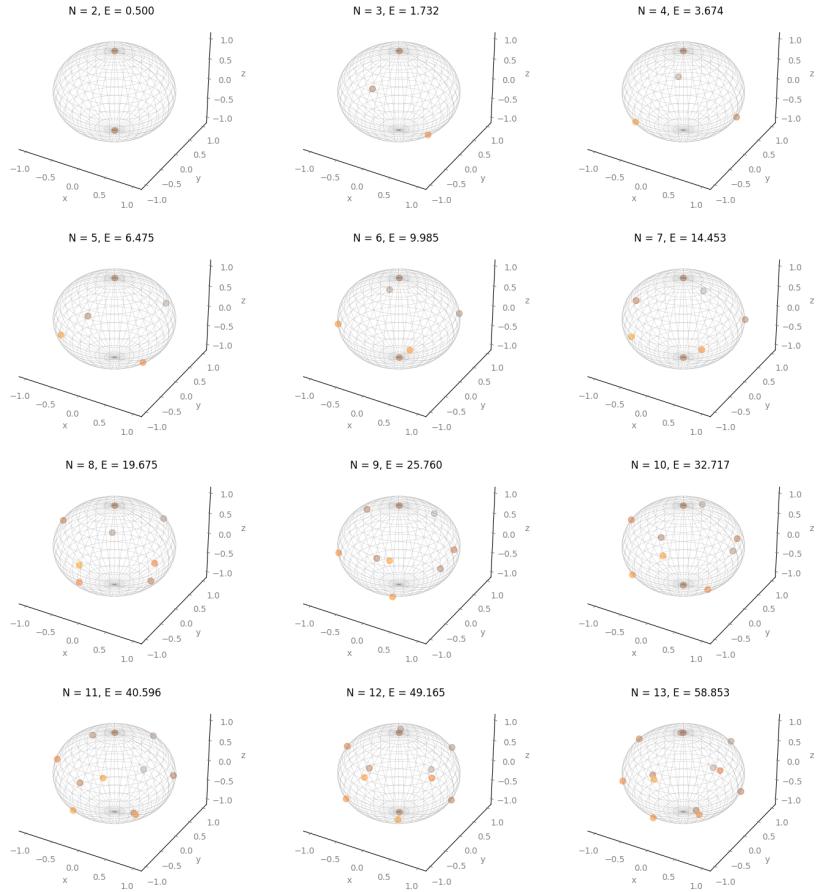


Slika 4: Brezdimenzijski dipolni moment rešitev, pridobljenih preko različnih metod, v odvisnosti od števila nabojev  $N$ . Rešitve so označene z manjšimi oranžnimi pikami, podatki iz Wikipedije [1] pa z večjimi zelenimi.

Na sliki 4 vidim, da se najbolje ujemata metodi BFGS in CG, najslabše pa Nelder-Mead in Powell. Ker je metoda BFGS hitrejša od metode CG, bom od tu dalje uporabljala le njo.

### 2.1.2 Rezultati

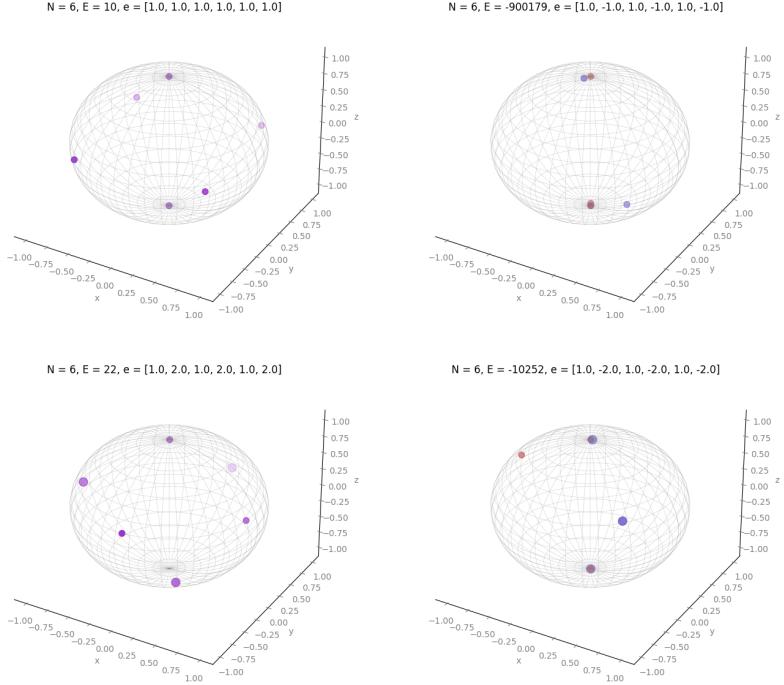
Pri Thomsonovemu problemu nas zanima razporeditev nabojev po krogli, ki je za število nabojev  $N$  od 2 do 13 prikazana na sliki 5, kjer so z bolj nasičeno in manj transparentno barvo narisani naboji, ki so bližje na 3D prikazu, tako dobimo malo boljši občutek za postavitev nabojev. Prikazana pa je tudi sfera na kateri so naboji razporejeni.



Slika 5: Rešitve Thomsonovega problema za število nabojev  $N$  od 2 do 13. Z bolj nasičeno in manj transparentno barvo narisani naboji, ki so bližje na 3D prikazu. S sivo je narisana sfera na kateri so naboji razporejeni. Rezultati so pridobljeni preko metode BFGS.

Na sliki 5 vidim, da sta naboja pri  $N = 2$  postavljena na premici, ki poteka skozi središče sfere, pri  $N = 3$  naboji sestavijo enakostranični trikotnik, za  $N = 4$  tetraeder, za  $N = 5$  tristrano bipiramido, itd. Ker je problem sferično simetričen, se pozicija nabojev spreminja (samo za kot  $\phi$ , saj je  $\theta$  fiksni zaradi pozicije prvega naboja  $\mathbf{r}_1 = (0, 0, 1)$ ). Na sliki so napisane tudi energije za posamezen  $N$ .

Zanimale so me tudi porazdelitve za različne naboje. Tako sem pogledala rešitve Thomsonovega problema za  $e_i \in \{1, -1, 1, -1, 1, -1\}$ ,  $e_i \in \{1, 2, 1, 2, 1, 2\}$  in  $e_i \in \{1, -2, 1, -2, 1, -2\}$  ter jih primerjala z porazdelitvijo, kjer so vsi naboji enaki  $e_i \in \{1, 1, 1, 1, 1, 1\}$ . Rezultati so prikazani na sliki 6



Slika 6: Rešitve Thomsonovega problema za število nabojev  $N = 6$  in različne naboje. Z manj transparentno barvo so narisani naboji, ki so bližje na 3D prikazu. Sivo je narisana sfera na kateri so naboji razporejeni. Rezultati so pridobljeni preko metode BFGS. Naboji, kjer imajo vsi enak predznak so narisani z vijolično, kjer pa imajo naboji različen predznak so eni narisani z rdečo, drugi pa z modro. Večji naboji so narisani z večjo piko.

Na sliki 6 so porazdelitve nabojev, kjer so vsi preznaki naboja enaki, dosegle globalni minimum, medtem ko ga porazdelitve z različno predznačenimi naboji niso. Pri različno predznačenih naboljih je porazdelitev in energija vsakič drugačna, program mi tudi sporoči, da se minimizacija ni izvedla uspešno zaradi deljenja z 0. To je smiselno, saj je energija pri členih z različno predznačenim naboljem minimalna takrat, ko sta naboja na istem mestu. V tej limiti gre  $|\mathbf{r}_i - \mathbf{r}_j| \rightarrow 0$  in s tem  $E \rightarrow -\infty$ . Tega pa program ne uspe izračunati, saj se v numeričnem postopku pojavi deljenje z 0, kar mi je tudi javil. Predvidevam, da bi pravilna porazdelitev naboja za različno preznačene naboje, kjer je negativno predznačenih, enako kot pozitivno predznačenih ( $N/2$ ), bila taka, da bi bili nabolji postavljeni tako kot so v rešitvi za istopredznačene za število naboljev  $N/2$ . Na teh mestih pa bi v isti točki nastopali negativno in pozitivno predznačeni nabolji.

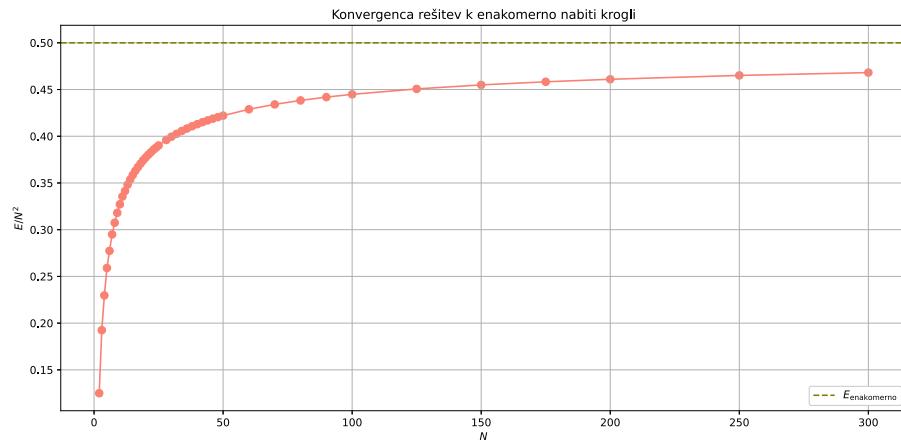
Nazaj se vračam na istopredznačene naboje  $e_i = 1$ . Tokrat me zanima, kako izgleda konvergenca k zvezni (enakomerni) porazdelitvi naboja. Energijo za enakometrno porazdelitev naboja po krogli izračunam kot

$$\tilde{E}_{\text{enakomerno}} = \frac{1}{2} \frac{e^2}{4\pi\epsilon_0 r},$$

kjer je v tem primeru  $e$  celoten naboj. Ker se v formuli (2) skupni naboj veča z večjim številom nabojev, moram zvezno limito opraviti na  $E/N^2$ , saj to poskrbi, da se v večanjem števila nabojev skupni naboj ne poveča

$$\lim_{N \rightarrow \infty} \frac{E}{N^2} = \frac{4\pi\epsilon_0 r}{e^2} \tilde{E}_{\text{enakomerno}} = E_{\text{enakomerno}} = \frac{1}{2}.$$

Rezultat je prikazan na sliki 7



Slika 7: Konvergenca  $E/N^2$  k zvezni porazdelitvi naboja. Rezultat je pridobljen preko metode BFGS. Z zeleno črtkano črto je prikazana zvezna limita.

Na sliki 7 vidim, da je konvergenca zelo počasna. Želela sem si dodati še točko  $N = 400$ , vendar je program potreboval preveč časa.

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Oddelek za fiziko

# Nelinearna minimizacija

Modelska analiza 1,  
Fizika, 2. stopnja, 1. letnik

Avtor: Kevin Jaksetič, vpisna številka: 28222050  
Mentorja: prof. dr. Simon Širca, doc. dr. Miha Mihovilovič

Študijsko leto 2022/2023

# Nalogi: Thomsonov problem, Optimalna vožnja skozi semafor

## Kazalo

<b>1 Thomsonov problem</b>	<b>2</b>
1.1 Rešitve ter nekateri najdeni lokalni minimumi . . . . .	3
1.2 Energija stabilnih ravovesnih stanj . . . . .	6
1.3 Primerjava natančnosti uporabljenih metod . . . . .	9
1.3.1 Metode z odvodi . . . . .	9
1.4 Primerjava časovnih zahtevnosti numeričnih metod . . . . .	11
1.5 Dodatno: enakostranični trikotnik iz nanožice . . . . .	12
<b>2 Optimalna vožnja skozi semafor</b>	<b>14</b>
2.1 Omejitev hitrosti: eksponentni člen . . . . .	15

# 1 Thomsonov problem

Na prevodno krogelno lupino nanesemo  $N$  enakih (klasičnih) nabojev. Zanima nas kako se razmestijo po površini. Seveda lahko za majhno število nabojev rešitve uganemo. Dva elektrona se bosta postavila čim dlje eden od drugega, trije se bodo postavili v oglišča navideznega enakostraničnega trikotnika in širje se bodo postavili v oglišča navideznega tetraedra. Za vse večja števila nabojev na površini postaja ugibanje vedno težje in zato prej ali slej nastopi numerično reševanje problema. Za to potrebujemo elektrostatsko energijo sistema zapisano kot funkcijo leg nabojev.

Želimo določiti minimum elektrostatske potencialne energije  $N$  elektronov, ki so vezani na krogelno lupino. V splošnem je elektrostatska energija med naboji enaka

$$E = \sum_i \sum_{j>i} \frac{e_i e_j}{4\pi\epsilon_0 |\mathbf{r}_i - \mathbf{r}_j|} , \quad (1)$$

kjer je  $e_i$   $i$ -ti naboj in  $\mathbf{r}_i$  njegov položaj. Zaenkrat se bomo osredotočili na primer, ko so vsi naboji enaki. V ustreznih enotah velja

$$E = \sum_i \sum_{j>i} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} . \quad (2)$$

Položaj  $i$ -tega naboja na enotski krogli je

$$\mathbf{r}_i = (\sin \theta_i \cos \varphi_i, \sin \theta_i \sin \varphi_i, \cos \theta_i) , \quad (3)$$

kjer sta  $\theta_i$  in  $\varphi_i$  polarni in azimutalni kot  $i$ -tega delca. Od tu sledi razdalja med dvema nabojem na krogelnih lupini

$$|\mathbf{r}_i - \mathbf{r}_j| = \sqrt{2 - 2 \sin \theta_i \sin \theta_j \cos(\varphi_i - \varphi_j) - 2 \cos \theta_i \cos \theta_j} . \quad (4)$$

Označimo koordinate posameznih nabojev s  $\mathbf{q}_i = (\theta_i, \varphi_i)$ . Za energijo je pomembno na kakšnem položaju so naboji glede na ostale naboje, ne pa tudi od tega kje točno se naboji nahajajo v našem koordinatnem sistemu. Zato lahko zaradi boljše preglednosti rešitev fiksiramo enega izmed nabojev na vrh sfere,  $\mathbf{q}_0 = (0, 0)$ . S tem tudi reduciramo  $E$  iz funkcije  $2N$  spremenljivk na funkcijo  $2(N - 1)$  spremenljivk. Energijo lahko končno zapišemo kot

$$E(\mathbf{Q}) = \sum_{i=1}^{N-1} \left( \frac{1}{\sqrt{2(1 - \cos \theta_i)}} + \sum_{j>i}^{N-1} \frac{1}{\sqrt{2 - 2 \sin \theta_i \sin \theta_j \cos(\varphi_i - \varphi_j) - 2 \cos \theta_i \cos \theta_j}} \right) , \quad (5)$$

kjer je  $\mathbf{Q} = (\theta_1, \theta_2, \dots, \theta_{N-1}, \varphi_1, \varphi_2, \dots, \varphi_{N-1})$ , prvi člen vsota parskih energij, ki vsebujejo fiksirani naboij, ter drugi člen vsota vseh ostalih parskih energij.

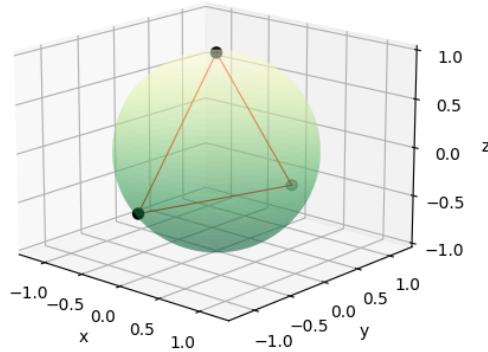
Problem sem numerično reševal s programskim jezikom *Python* s funkcijo `minimize` iz modula `scipy.optimize`. Tej funkciji lahko ukažemo s katero od podprtih metod naj izračuna rešitev. Najprej bom predstavil rešitve, ki sem jih dobil z metodo SLSQP, nato pa bom še primerjal preizkušene metode.

## 1.1 Rešitve ter nekateri najdeni lokalni minimumi

Programu moramo podati tudi začetni približek oziroma začetno ugibanje rešitve. Preprosto se lahko delamo, da o rešitvah ne vemo nič in za začetno ugibanje izberemo naključne točke. Polarne kote posameznih nabojev žrebamo iz intervala  $[0, \pi]$ , medtem ko azimutne kote žrebamo iz intervala  $[0, 2\pi)$ .

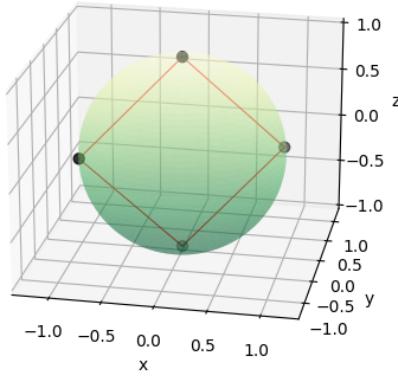
Pomembno je omeniti, da z minimizacijsko metodo lahko najdemo lokalni minimum namesto globalnega. To pomeni, da smo našli labilno ravnovesje namesto stabilnega. Poglejmo si prikaze najdenih ravnovesij obeh vrst za nekaj različnih  $N$ -jev.

Najdeno ravnovesje za 3 naboje



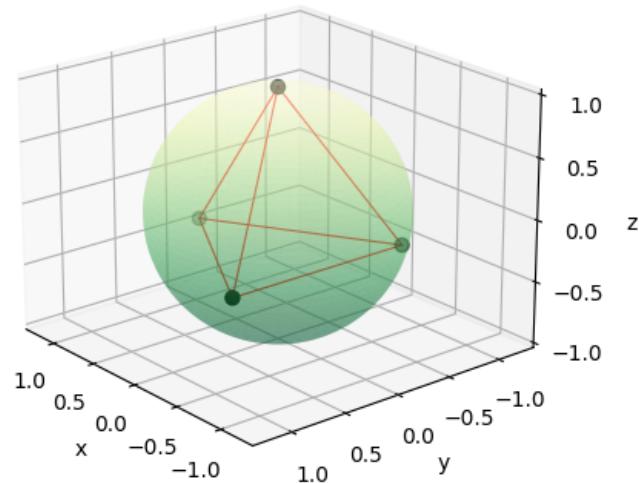
Slika 1: Stabilno ravnovesno stanje za tri enake naboje. Ti se postavijo v oglišča enakostraničnega trikotnika.

Najdeno ravnovesje za 4 naboje



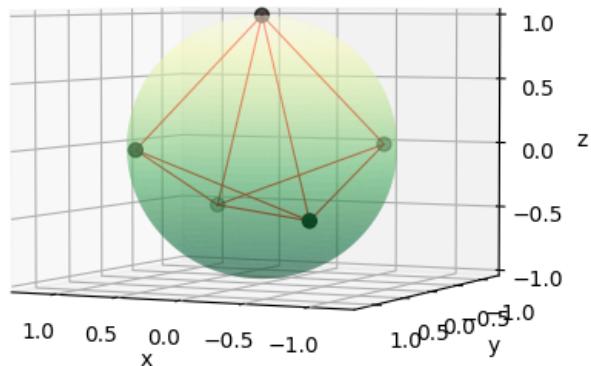
Slika 2: Labilno ravnovesno stanje za 4 enake naboje. Ti lahko mirujejo v ogliščih kvadrata.

Najdeno ravnovesje za 4 naboje



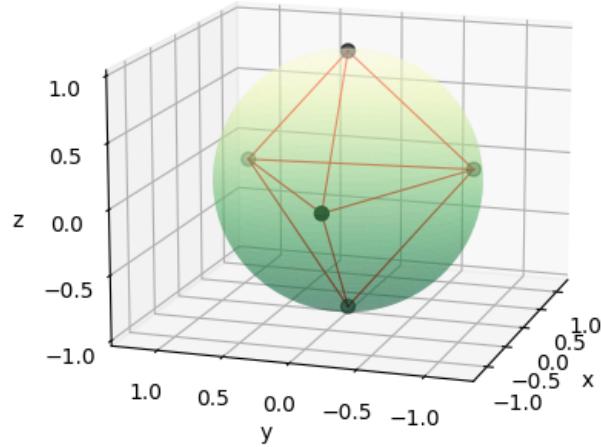
Slika 3: Stabilno ravnovesno stanje za 4 enake naboje. Ti se postavijo v oglišča tetraedra.

Najdeno ravnovesje za 5 nabojev



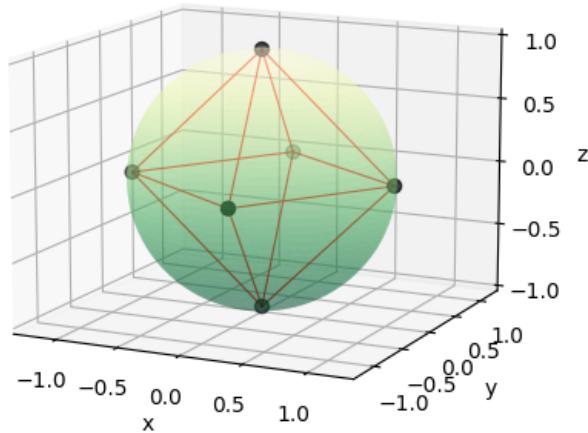
Slika 4: Labilno ravnovesno stanje za 5 enakih nabojev. Naletel sem tudi na stanje, pri katerem se postavijo v štiristrano piramido.

### Najdeno ravnovesje za 5 nabojev



Slika 5: Stabilno ravnovesno stanje za 5 enakih nabojev. Ti se postavijo v oglišča tristrane bipiramide.

### Najdeno ravnovesje za 6 nabojev

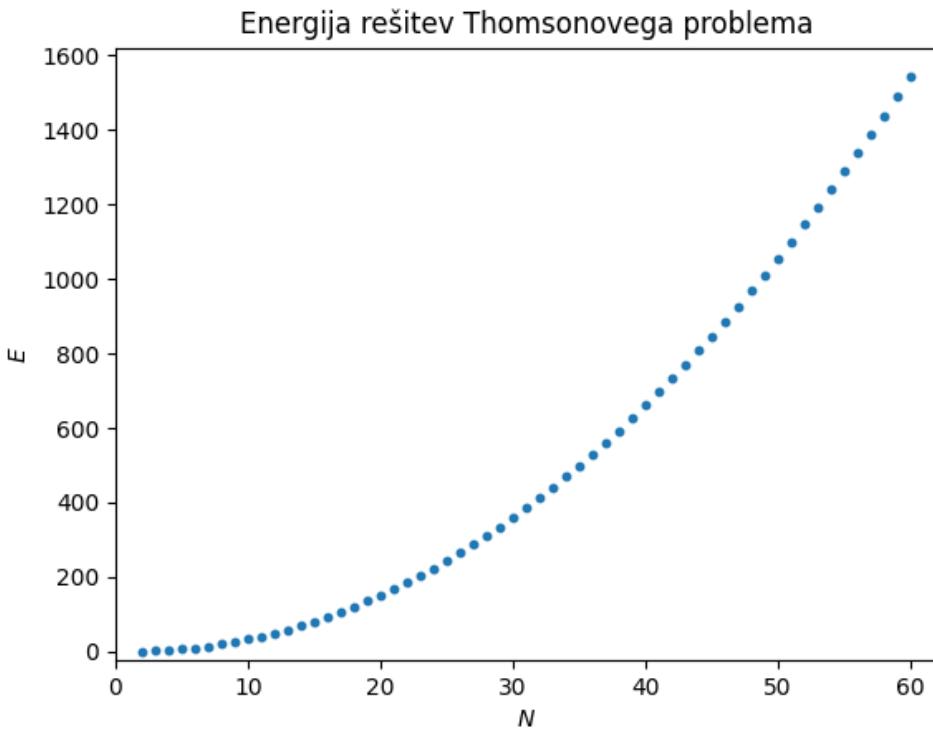


Slika 6: Stabilno ravnovesno stanje za 6 enakih nabojev. Ti se postavijo v oglišča štiristrane bipiramide.

Za lažjo predstavo sem naredil tudi nekaj animacij, kjer se krogla vrati.

## 1.2 Energija stabilnih ravnovesnih stanj

Najprej si poglejmo energijo rešitev (Slika 7), ki so tabelirane na spletni strani *Wikipedia*. Z dodajanjem nabojev je minimalna energija sistema vedno večja, kar ni nobeno presenečenje. Zaradi narave elektrostatskega potenciala med dvema nabojem ( $V \propto 1/r$ ) je funkcija  $E(N)$  tudi konveksna (kot pričakovano). Odvisnost zgleda kvadratična, kar bomo kasneje pokazali da do neke mere drži.



Slika 7: Tabelirane energije rešitev Thomsonovega problema za različna števila nabojev na krogelni lupini. Graf spominja na kvadratično funkcijo, kar bomo kasneje videli da približno drži. Na grafu je enota dvakratna vrednost energije dveh nabojev v ravnovesju.

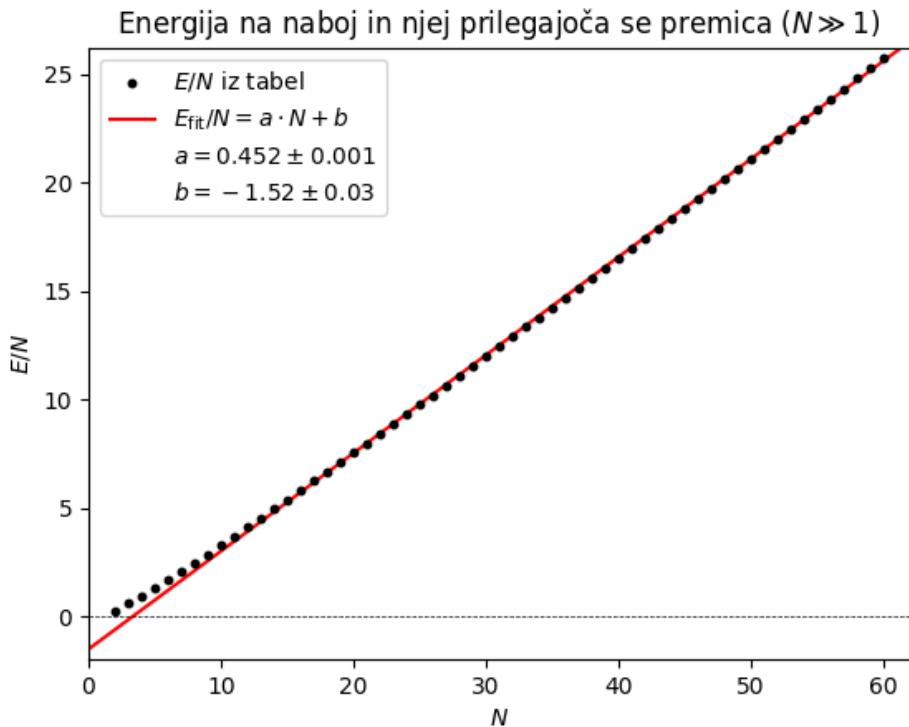
Smiselno je preučiti tudi energijo na nabo,  $E/N$ . Ta seveda narašča s povečevanjem števila nabojev. Izkazalo se je, da se pri velikih  $N$  ta funkcija obnaša skoraj linearno. Z linearno regresijo sem določil premico, ki se najbolje prilega tabeliranim vrednostim  $E/N$ , pri čemer sem upošteval le podatke za  $N > 10$ . To sem storil zaradi očitno nelinearnega obnašanja funkcije pri majhnih  $N$ . Podatki in njihova linearna regresija so prikazani na Sliki 8. Linearno

Zanimivo je še pogledati kolikšna je povprečna energija Coulombskega odboja med dvema nabojem,  $\langle E_{ij} \rangle = E/p$ , kjer je  $p$  število parov in ga izračunamo kot

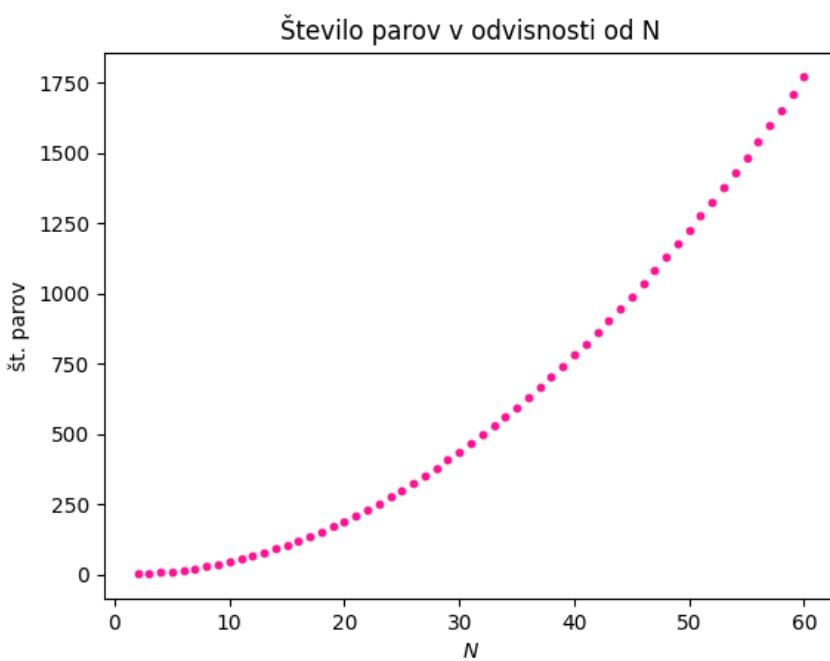
$$p = \frac{N(N - 1)}{2} .$$

Velja torej

$$\langle E_{ij} \rangle = \frac{2E}{N(N - 1)} . \quad (6)$$

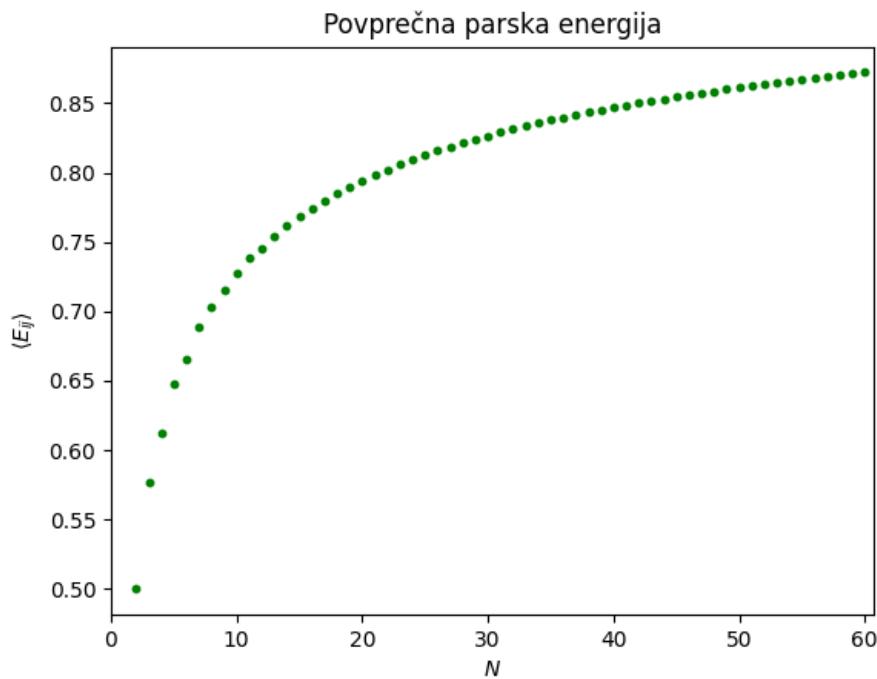


Slika 8: Tabelirane vrednosti  $E/N$  in njim najboljše prilegajoča se premica.



Slika 9: Število parov nabojev  $p$  v odvisnosti od  $N$ . Velja  $\frac{N(N-1)}{2}$ , kar je podobno kot  $E(N)$  nekakšna parabola.

Pri enačbi (6) torej delimo dve paraboli. Na Sliki 10 je prikazan rezultat tega deljenja.

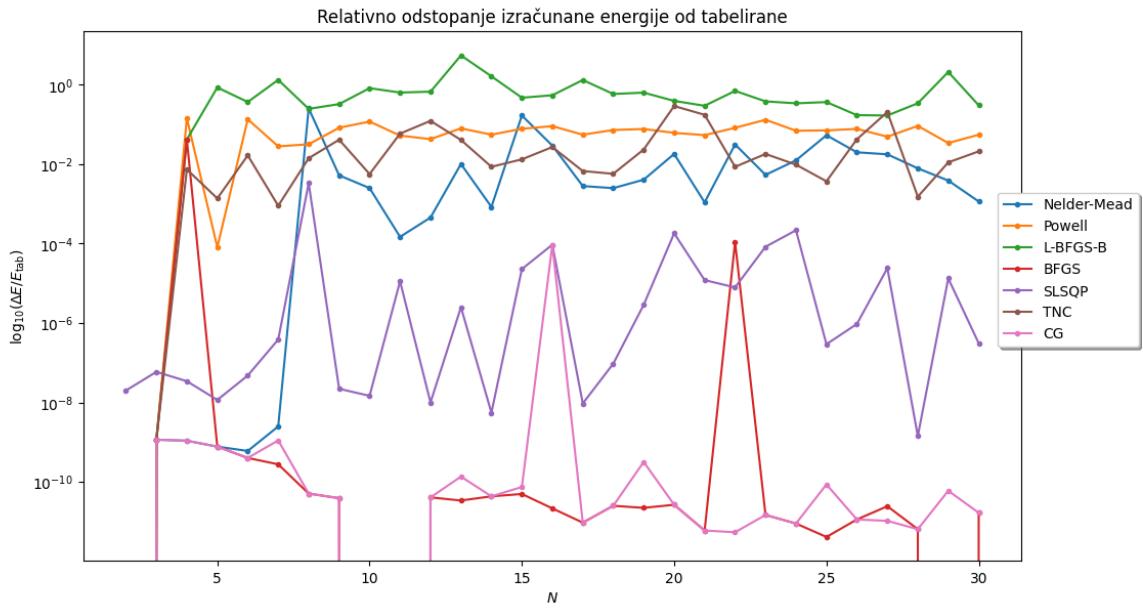


Slika 10: Povprečna parska energija v odvisnosti od števila nabojev. Ker se razdalje med naboji z večanjem njihovega števila seveda manjšajo, se s tem veča povprečna energija na par.

Pričakovano je  $\langle E_{ij} \rangle$  naraščajoča funkcija  $N$ . Manjše razdalje med naboji vodijo v večje parske energije.

### 1.3 Primerjava natančnosti uporabljenih metod

Vse metode nimajo iste natančnosti. Pa jo preučimo. Kot mero za natančnost sem vzel odstopanje izračunane energije od tabelirane energije iz Wikipedije. Seveda bi lahko obravnavali tudi odstopanja izračunanih leg delcev od pravilnih, vendar nam povzroča težave to, da je problem rotacijsko invarianten. Tako bi lahko na primer gledali odstopanje vsote razdalj med naboji.



Slika 11: Relativno odstopanje izračunane energije od tabelirane. Pri tem nismo izključili, da je algoritem lahko našel lokalni minimum namesto globalnega. V tem primeru je napaka seveda velika.

#### 1.3.1 Metode z odvodi

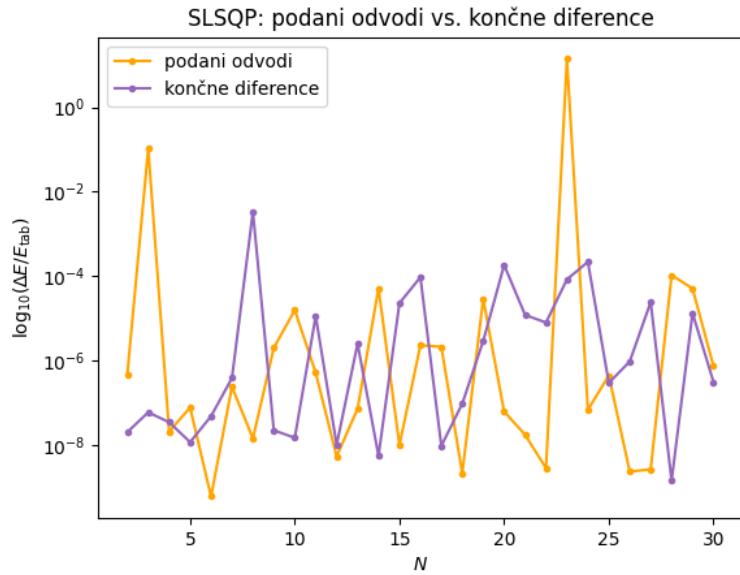
Kar nekaj uporabljenih algoritmov za izračun minimuma funkcije uporablja tudi odvode. Če mu jih ne podamo, jih aproksimira s končnimi diferencami. Tukaj se lahko skriva znatna napaka. Zanimivo bi bilo pogledati izboljšanje natančnosti ob podanih odvodih. Takšne metode so CG, BFGS, L-BFGS-B, SLSQP in TNC.

Najprej moramo izračunati še odvode energije. To so

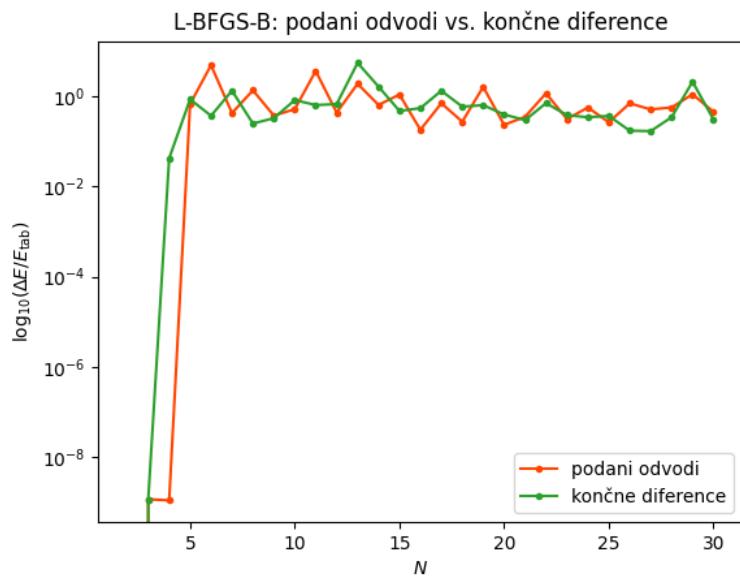
$$\frac{\partial E}{\partial \theta_i} = \sum_{j, j \neq i} \frac{\cos(\varphi_i - \varphi_j) \cos \theta_i \sin \theta_j - \cos \theta_j \sin \theta_i}{(2 - 2 \sin \theta_i \sin \theta_j \cos(\varphi_i - \varphi_j) - 2 \cos \theta_i \cos \theta_j)^{3/2}}, \quad (7)$$

$$\frac{\partial E}{\partial \varphi_i} = \sum_{j, j \neq i} -\frac{\sin(\varphi_i - \varphi_j) \sin \theta_i \sin \theta_j}{(2 - 2 \sin \theta_i \sin \theta_j \cos(\varphi_i - \varphi_j) - 2 \cos \theta_i \cos \theta_j)^{3/2}}. \quad (8)$$

Za metodi SLSQP in L-BFGS-B sem dobil slednje rezultate.



Slika 12: Odstopanje od prave energije glede na to ali smo metodi SLSQP podali odvode ali ne. Vidimo, da rezultat ni nujno boljši.

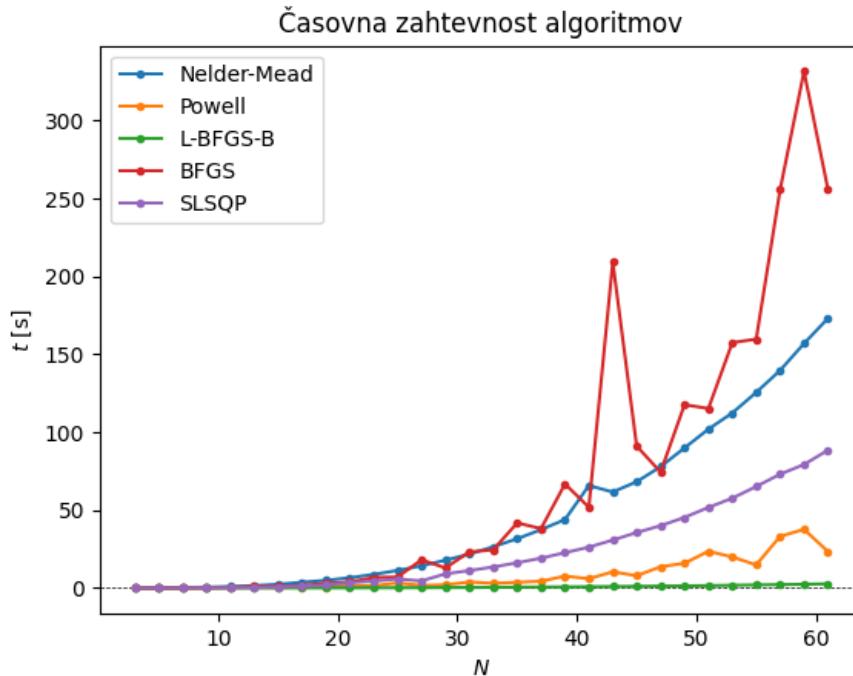


Slika 13: Odstopanje od prave energije glede na to ali smo metodi L-BFGS-B podali odvode ali ne. Vidimo, da rezultat ni nujno boljši.

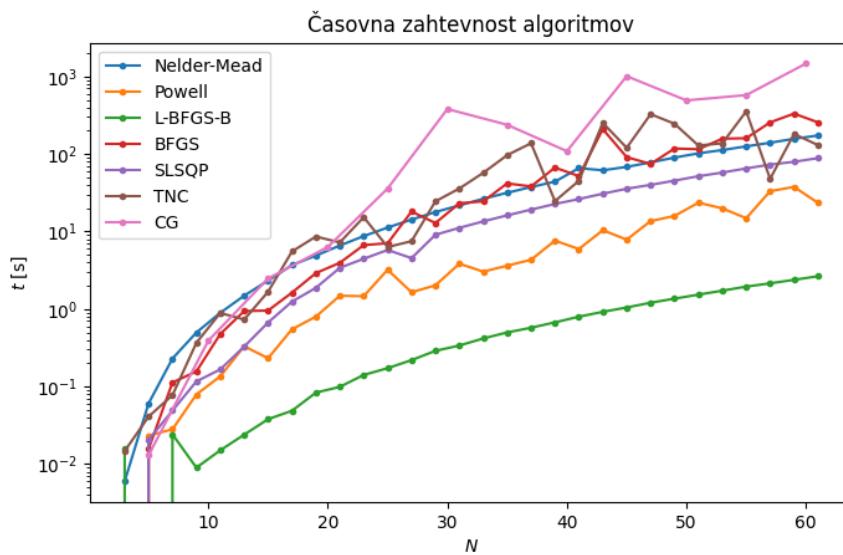
Sklepam, da mogoče moje začetno ugibanje rešitve, ki ga podam metodi, ni optimalno in zaradi tega algoritmi dostikrat najdejo lokalni minimum namesto globalnega. Komentar: če ne bi poznali rešitve problema, bi lahko večkrat pognali metodo in med sabo primerjali njene rezultate, da bi s tem dobili idejo koliko je zanesljiva.

## 1.4 Primerjava časovnih zahtevnosti numeričnih metod

Poleg željene natančnosti je pri metodah pomembna časovna zahtevnost algoritma. Pomeril sem računske čase nekaj različnih metod, pri čemer algoritmom nisem podal odvodov.



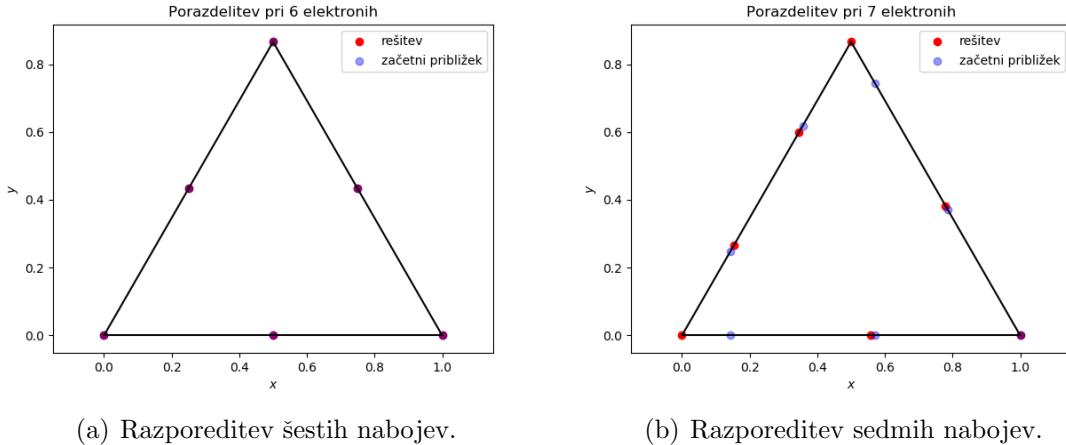
Slika 14: Časovna zahtevnost nekaterih uporabljenih metod.



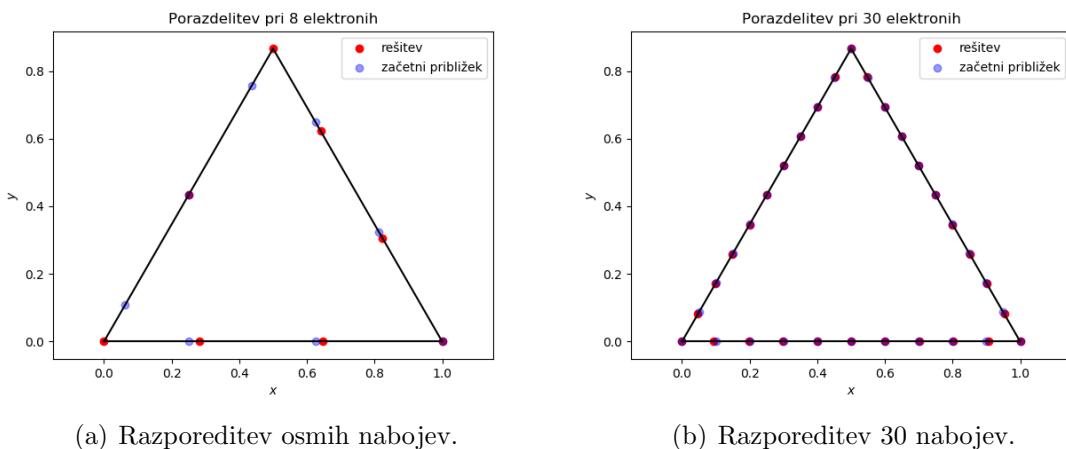
Slika 15: Časovna zahtevnost uporabljenih metod, v logaritemski skali.

## 1.5 Dodatno: enakostranični trikotnik iz nanožice

V tem primeru minimiziramo funkcijo samo ene spremenljivke, naravnega parametra  $s$ , ki teče vzdolž stranic trikotnika in sicer od 0 do 3. Ta primer sem si izbral, ker je bila to moja zaključna naloga pri predmetu Matematična-fizika I na 1. stopnji študija fizike. Ker nam je postopek že znan, bom samo prikazal rezultate.



Slika 16: Razporeditev nabojev je skoraj ekvidistantna (glej modre pike).

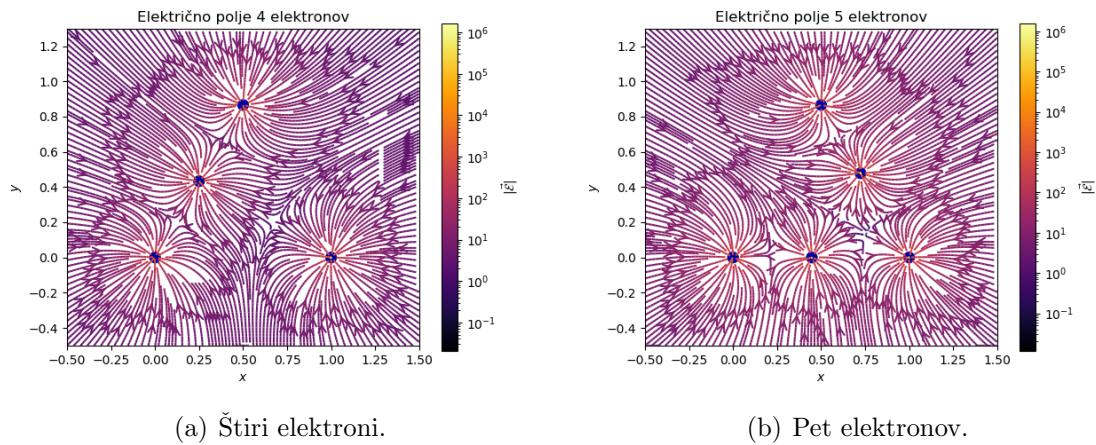


Slika 17: Razporeditev nabojev je skoraj ekvidistantna (glej modre pike).

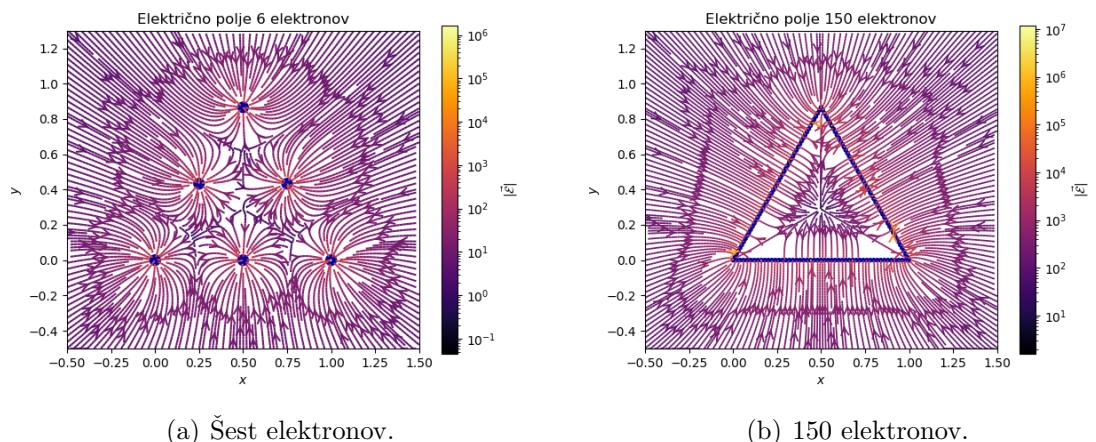
Porazdelitev nabojev na enakostraničnem trikotniku je skoraj ekvidistantna in vedno bolj stremi k enakomerni nabitosti, ko večamo število nabojev na žici. Če poznamo porazdelitev nabojev, lahko izračunamo tudi polje kot

$$\vec{\mathcal{E}}(\vec{r}) = \sum_{i=1}^N \frac{\vec{r}_i - \vec{r}}{|\vec{r}_i - \vec{r}|^3}, \quad \vec{r} \neq \vec{r}_i. \quad (9)$$

Poglejmo si električno polje v ravnini trikotnega okvira.



Slika 18: Električno polje v ravnini trikotnega okvira.



Slika 19: Električno polje v ravnini trikotnega okvira.

Komentar: kaj podobnega bi bilo v 3d težko prikazati.

# **Modelska analiza 1**

# **Numerična minimizacija**

3. naloga

Črtomir Perharič  
Vpisna številka: 28222056  
Asistent: doc. dr. Miha Mihovilovič

25. oktober 2022

# 1 Zastavitev problema

Rešujemo nelinearni optimizacijski problem, kjer minimiziramo funkcijo več spremenljivk. Takšne probleme rešujemo z numerično minimizacijo. Različne metode le-te smo spoznali na predavanjih.

Najprej bomo rešili Thomsonov problem, kjer na prevodno kroglo nanesemo N enaki nabojev in opazujemo, kako se naboji razporedijo po sferi. Nalogo rešimo z minimizacijo elektrostatske energije

$$\sum_{i,j>i}^N \frac{e^2}{4\pi\epsilon_0 \|\mathbf{r}_i - \mathbf{r}_j\|} = \min. \quad (1)$$

To enačbo lahko prepišemo v brezdimenzijsko obliko, kjer postavimo  $4\pi\epsilon_0 \rightarrow 1$  in naboje kvantiziramo v večkratnikih osnovnega naboja. Enačbo za brezdimenzijsko energijo  $\epsilon$  je potem enaka

$$\epsilon = \sum_{i,j>i}^N \frac{e^2}{\|\mathbf{r}_i - \mathbf{r}_j\|}, \quad (2)$$

kjer smo zaradi kompaktnosti pisali  $e_0 \rightarrow e$ .

V drugem delu naloge pa bomo reševali minimizacijo akcije modela vožnje skozi semafor, ki smo jo spoznali pri prvi nalogi. Enačbe je potrebno prepisati v diskretno obliko, integrale pa izračunati s trapezno formulo:

$$F = \left[ \frac{1}{2} \left( \frac{v_0 - v_{-1}}{\Delta t} \right)^2 + \left( \frac{v_1 - v_0}{\Delta t} \right)^2 + \dots + \frac{1}{2} \left( \frac{v_n - v_{n-1}}{\Delta t} \right)^2 \right] \Delta t.$$

V diskretno obliko je potrebno prepisati še pogoj  $\int_0^1 v dt = 1$ , kar zapišemo kot

$$\left( \frac{1}{2}v_0 + v_1 + \dots + \frac{1}{2}v_n \right) \Delta t \leq 1.$$

Definiramo novi funkcional

$$\tilde{F} = F + F_1,$$

kjer je

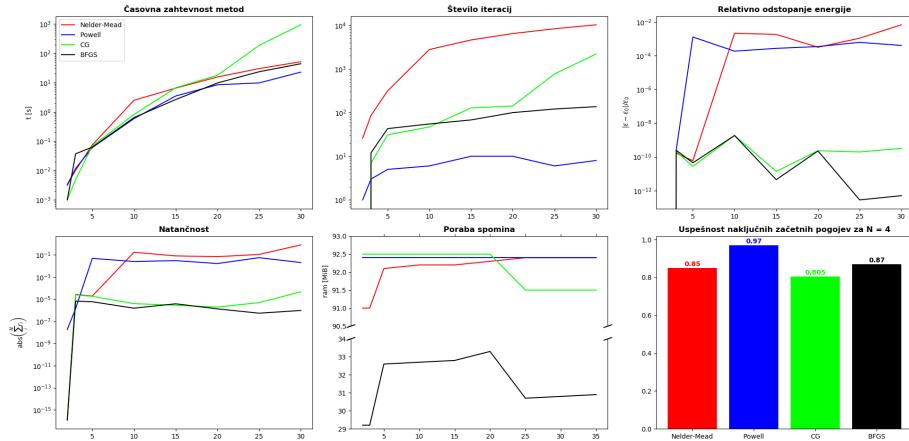
$$F_1 = e^{\kappa(\frac{1}{2}v_0 + v_1 + \dots + \frac{1}{2}v_n - 1/\Delta t)}.$$

Funkcional  $F_1$  bo imel majhno vrednost, ko bo veljala neenakost v pogoju za prepotovano pot, če bo  $\kappa > 0$ . Minimiziramo torej funkcional  $\tilde{F}$ , kjer vpliv  $F_1$  določamo s vrednostjo  $\kappa$ .

# 2 Thomsonov problem

## 2.1 Preučevanje metod

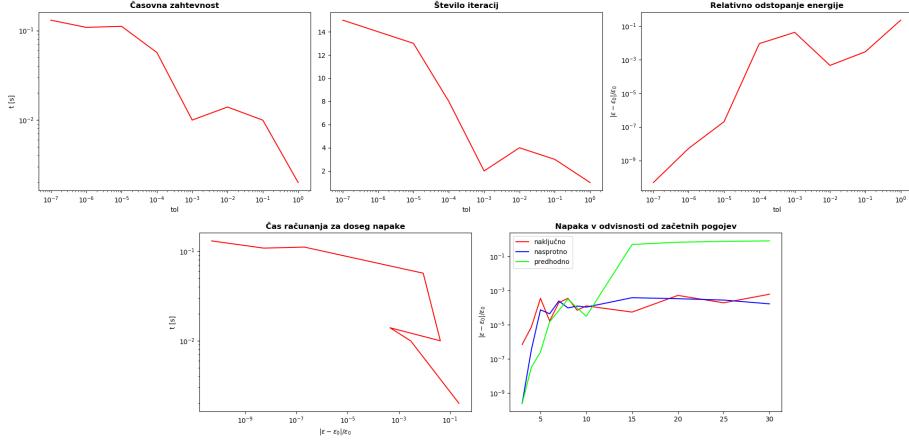
Najprej je smiselno preučini različne računske zahtevnosti posameznih metod numerične minimizacije. Ugotovitve prikazuje slika 1.



Slika 1: Računske zahtevnosti posameznih metod. Na zgornjih grafih so prikazani časovna zahtevnost (levo), število iteracij (sredina) in relativno odstopanje energije (desno). V spodnji vrsti pa je prikazano odstopanje dolžine vsote radij vektorjev nabojev kot še eno merilo za natančnost (levo), poraba spomina (sredina) in uspešnost minimizacije pri povsek naključnih začetnih pogojih za 200 poskusov (desno).

Ugotovimo, da je najhitrejša metoda *Powell*, najpočasnejša pa metoda *CG*. Približno skladno s tem je odvisno tudi število iteracij, čeprav vidimo, da največ ponovitev opravi metoda *Nelder Mead*. Najboljši približek dejanskim potencialom (kot jih podaja wikipedia) izračuna metoda *BFGS*, kar potrjuje tudi levi spodnji graf, kjer primerjamo razporeditve elektronov. Poraba spomina očitno ni odvisna od števila nabojev, je pa očitno manjša pri metodi *BFGS*. Na desnem spodnjem grafu je prikazana še uspešnost minimizacije pri 200 poskusih z naključnimi začetnimi pogoji. Vidimo, da je najbolj uspešna metoda *Powell*, najmanj pa *CG*. Vendar je potrebno poudariti, da tej metodi nisem podal odvodov, zato jih je izračunala sama z diferencami. Verjetno bi bila uspešnost metode ob predpisanih odvodih boljša.

Ker je metoda *Powell* najhitrejša, hkrati pa tudi dosti natančna, sem se odločil, da jo uporabim v nadaljnih poskusih. Funckiji `scipy.optimize.minimize` lahko predpišemo mero za željeno natančnost *tol*. Pogledal sem še, kako vrednost tega parametra vpliva na računske zahtevnosti metode (slika 2).



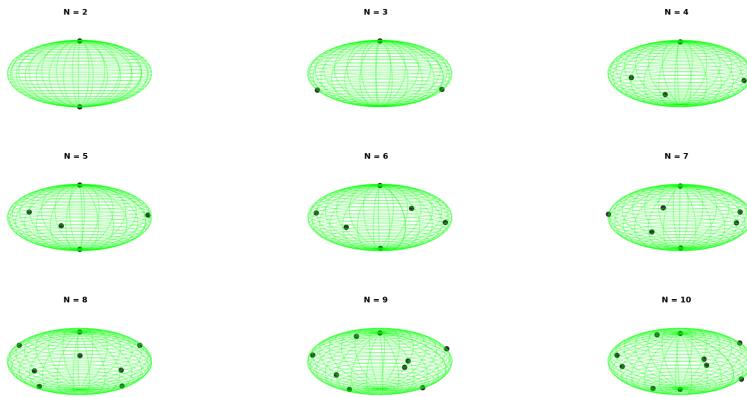
Slika 2: Računske zahtevnosti metode Powell v odvisnosti od parametra  $tol$  za  $N = 4$  in vpliv izbiranja začetnih pogojev na energijo za (desno spodaj).

Ugotovimo, da se čas računanja in število iteracij z manjšanjem parametra povečujeta, napaka pa pada. To je seveda pričakovano. Opazimo, da lahko dosežemo tudi dosti manjše napake energije  $\sim 10^{-10}$ . Levi spodnji graf nam prikazuje morda numerično najbolj pomemben podatek: časovno odvisnost od velikosti napake.

Pomembno vprašanje je, kako pametno izbrati začetne pogoje, da bodo odstopanja čim manjša. Odločil sem se preizkusiti 3 načine: naključno, kjer vse naboje naključno razporedimo po krogelni lupini, nasprotno, kjer vsak drugi naboj postavimo nasproti predhodnika in predhodno, kjer rešitvi za  $N-1$  nabojev naključno dodamo še  $N$ -ti naboj. Rezultate prikazuje levi spodnji graf na sliki 2. Opazimo, da za večje  $N$  predhodna metoda nekoliko slabše deluje od drugih dveh in vrne tudi rešitve s 10% napako. V nadaljevanju sem uporabljal metodo nasprotnih nabojev.

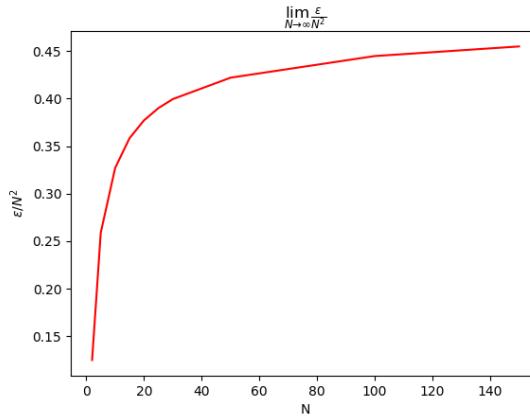
## 2.2 Preučevanje rešitev

Slika 3 prikazuje grafično razporeditev  $N$  nabojev Thomsonovega problema.



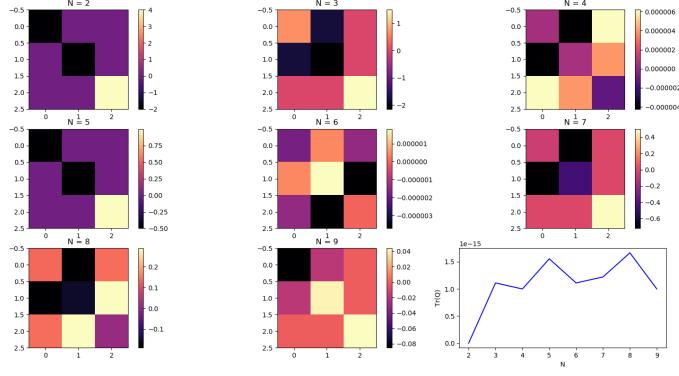
Slika 3: Grafični prikaz razporeditve nabojev za prvi N nabojev. Poudarimo, da so naboji razporejeni po krogelni lupini in ne po elipsoidu, čeprav na prvi pogled izgleda obratno, saj so razmerja višin in širin na grafih različni od 1.

**Limitni primer:** Zanimivo je pogledati, kaj se dogaja z elektrostatsko energijo ko  $N \rightarrow \infty$ . Takrat se bližamo primeru zvezne ploskovne porazdelitve naboja  $\sigma$ . Po naši definiciji energije bi pričakovali, da  $\lim_{N \rightarrow \infty} \epsilon/N^2 = 1/2$ , kar potrdi graf na sliki 4. Opazimo, da energija res konvergira, čeprav zelo počasi, proti vrednosti  $\frac{1}{2}$ .



Slika 4: Približevanje kontinumski sliki nabite krogelne lupine

**Kvadrupol:** Izračunamo lahko tudi komponente tenzorja kvadrupolnega



Slika 5: Prikaz kvadrupolnih momentov in spremenjanje njihove sledi v odvisnosti od  $N$  (desno spodaj), kot mera za natančnost metode. Za boljšo natančnost je bila uporabljena metoda *BFGS*.

momenta

$$Q_{ij} = \sum_{ij}^N (3r_i^k r_j^k - |\mathbf{r}^k| \delta_{ij}),$$

ki jih prikazuje slika 5. Zadnji spodnji graf na sliki 5 nam prikazuje spremenjanje sledi tenzorja v odvisnosti od števila nabojev  $N$ . Po definiciji je ta tenzor brezsleden, zato lahko odstopanja prav tako obravnavamo kot mero za napako metode.

**Različni naboji:** Kot zanimivost lahko na sliki 6 opazujemo še razporeditev nabojev za nključnimi velikostmi  $e \in 1, 2$ .

$$\mathbf{N = 10}, \varepsilon = 86.5, \sum_i e_i = 17 \quad \mathbf{N = 12}, \varepsilon = 93.4, \sum_i e_i = 17$$



$$\mathbf{N = 14}, \varepsilon = 134.4, \sum_i e_i = 21 \quad \mathbf{N = 16}, \varepsilon = 267.7, \sum_i e_i = 26$$



Slika 6: Grafični prikaz razporeditve različno velikih nabojev. S črno barvo so prikazani naboji z velikostjo 1, z rdečo pa naboji z velikostjo 2.

Opazimo, da se tudi tu energija povečuje z večanjem  $N$ , če je vsota nabojev enaka. Hkrati pa opazimo, da ravno ta vsota dosti bolj vpliva na vrednost energije.

### 3 Model vožnje skozi semafor

Ker sem imel težave z velikimi vrednostmi  $\kappa$ , sem funkcional  $F_1$  preuredil v obliko

$$F_1 = e^{\kappa(\dots)} + e^{-\kappa(\dots)}.$$

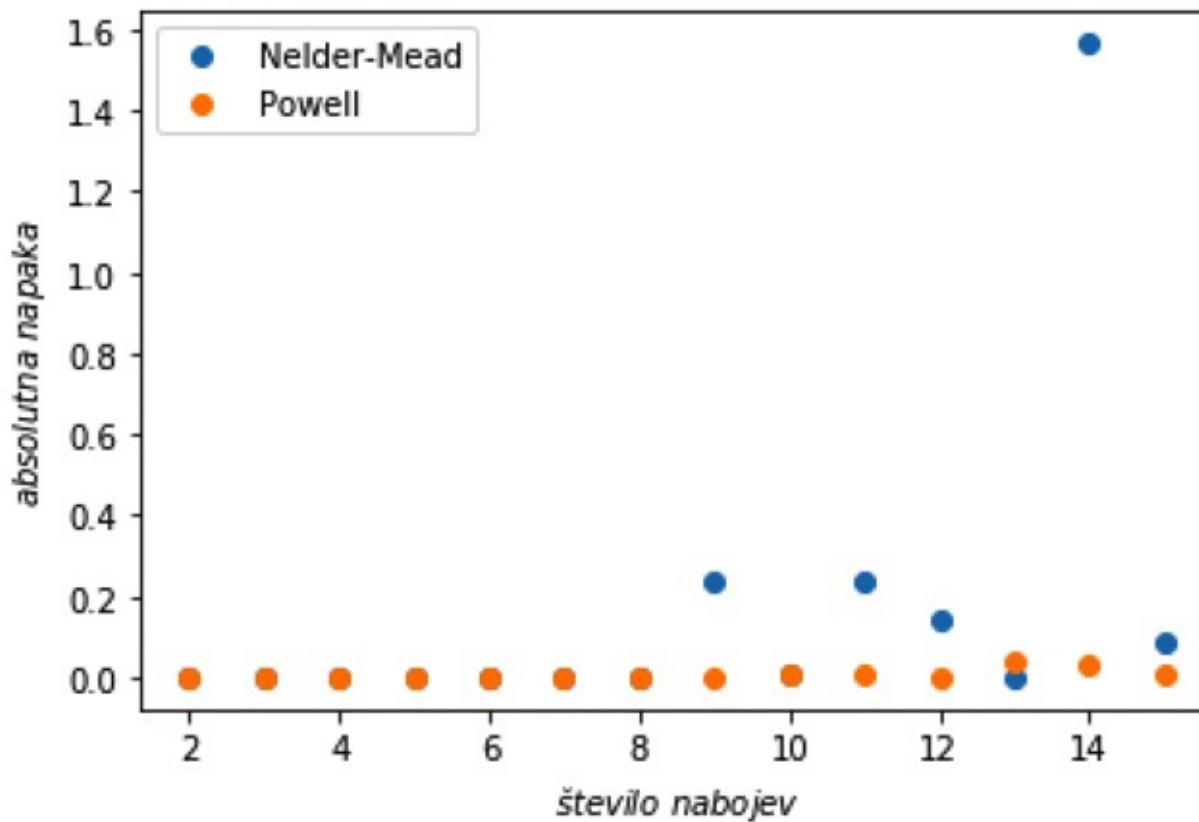
Za upoštevanje pogoja začetne hitrosti pa sem funkcionalu dodal še člen

$$F_2 = e^{\kappa'(v_{n=0} - v_0)} + e^{-\kappa'(v_{n=0} - v_0)}.$$

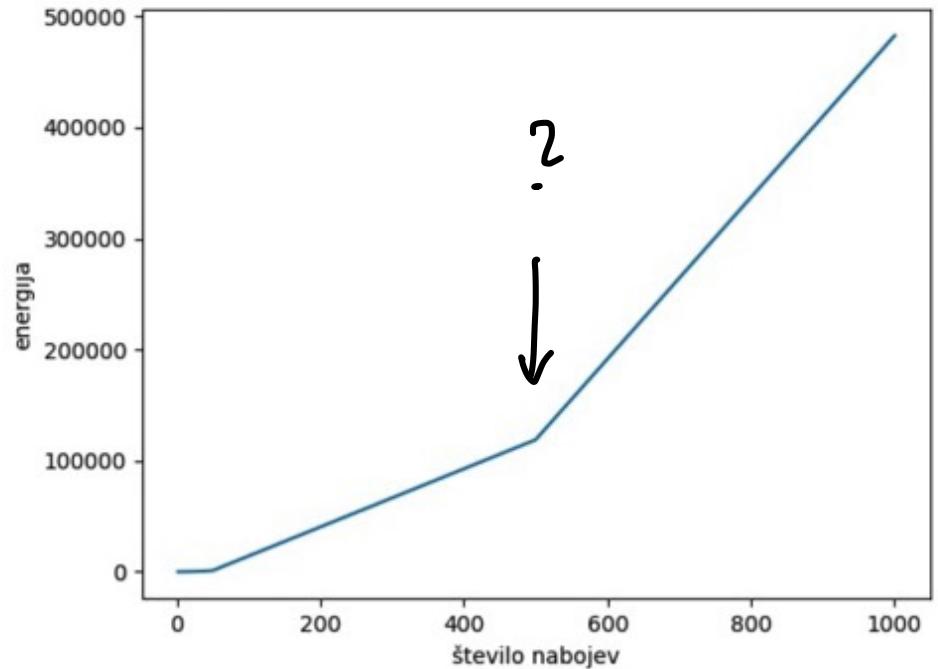
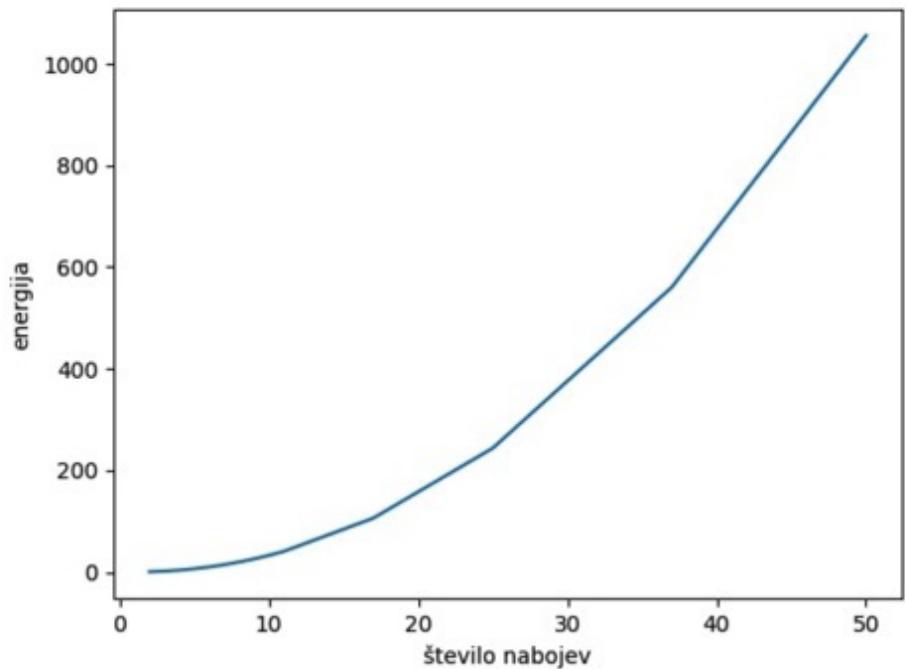
#### 3.1 Prost robni pogoj

Za minimizacijo sem uporabil metodo *BFGS*. Rešitve in njihova odstopanja za različne velikosti  $\kappa$  prikazuje slika 7.

2.



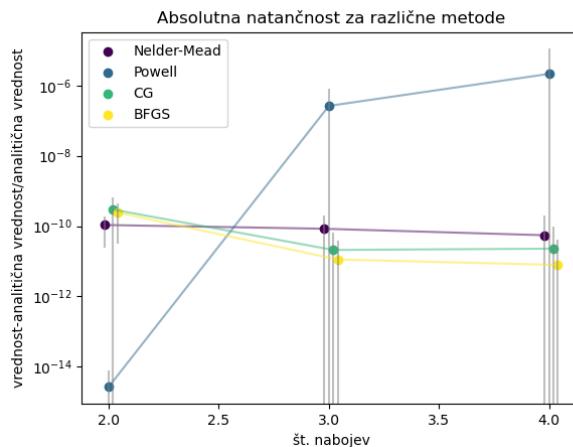
Slika 1: Absolutna napaka pri minimizaciji elektrostatske energije  $N$  enakih točkastih nabojev pri uporabi dveh različnih minimizacijskih metod. Napaka je izračunana glede na vrednosti iz Wikipedije ([https://en.wikipedia.org/wiki/Thomson\\_problem](https://en.wikipedia.org/wiki/Thomson_problem)).



Slika 2: Velikost potencialne energije za  $N = 50$  na desni, ter za  $N = 1000$ . V grobem lahko iz grafov vidimo, da se potencialna energija narašča eksponentno s številom nabojev.

# Emca Mlinar

Seveda pa standardna deviacija ni nujno dobro merilo, lahko bi se namreč zgodilo, da bi metoda sistematično odstopala od prave vrednosti. Ker imamo analitično rešitev Thomsonovega problema za  $N = 2, 3, 4$  lahko preverimo, kako je z absolutno napako. Te napake so prikazane na sliki 4.

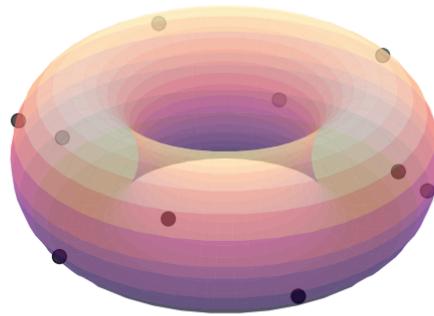


Slika 4: Preverjanje točnosti metod za manjše število nabojev, ko imamo analitično rešitev. Sivi intervali predstavljajo standardno deviacijo poskusov (merilo, ki smo ga prej uporabili za določanje natančnosti).

Opazimo lahko, da je prava (analitična) vrednost v okviru napake, ki jo določimo statistično (razen za primer  $N = 2$ , ki pa je očitno premajhno število nabojev), torej je merilo od prej dobro merilo za natančnost.

## 3 Torus

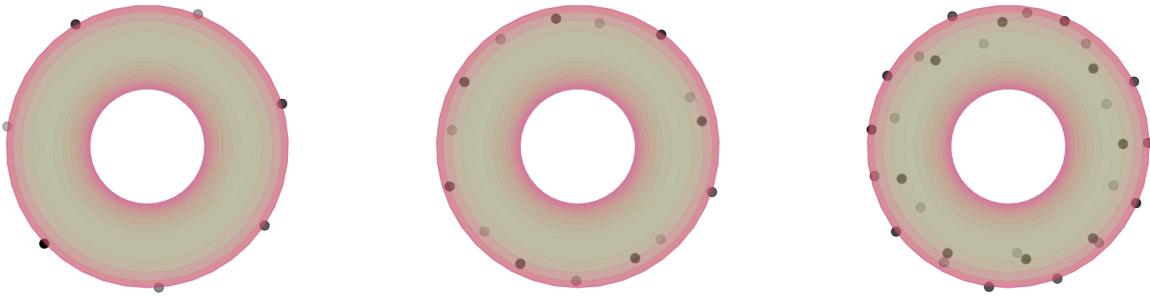
Poglejmo si sedaj primer malo manj (a še vedno zelo) simetričnega objekta. Izbrala sem si torus, primer razporeditve nabojev na torusu lahko vidimo na sliki 5



Slika 5: Primer razporeditve 10 nabojev na torusu.

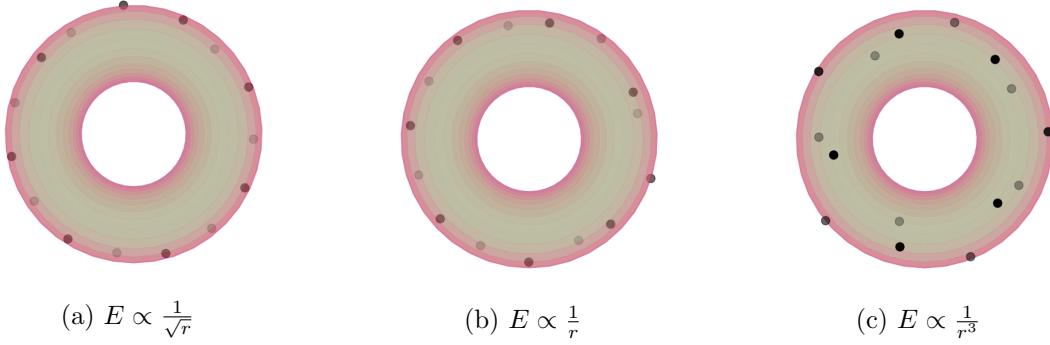
Razporeditev izgleda precej pričakovano, a če ga pogledamo od zgoraj vidimo, da je močno preferiran zunanji rob torusa (glej sliko 6).

Sklepam, da se to zgodi, ker bi naboj na sredini vplival po malem na vse naboje, če pa je na robu pa na oddaljene skoraj ne vpliva. Torej, z drugimi besedami, da je to posledica dosega potenciala.



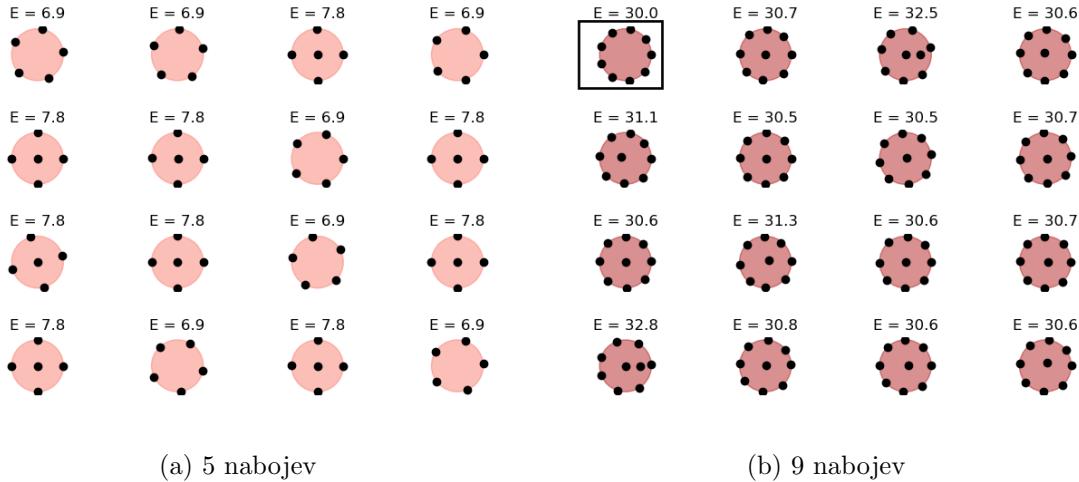
Slika 6: Pogled na razporeditev za 7, 15 in 30 nabojev na torusu od zgoraj.

Preverimo. Na sliki 7 je prikazana razporeditev nabojev na torusu za različne odvisnosti potenciala od razdalje med naboji.



Slika 7: Pogled na torus s 15 naboji iz ptičje perspektive za različne potence potenciala.

Vidimo, da so naboji bolj enakomerno razporejeni po celotnem torusu za višje potence, medtem ko so za potence 1 ali nižje naboji praktično samo na robu.

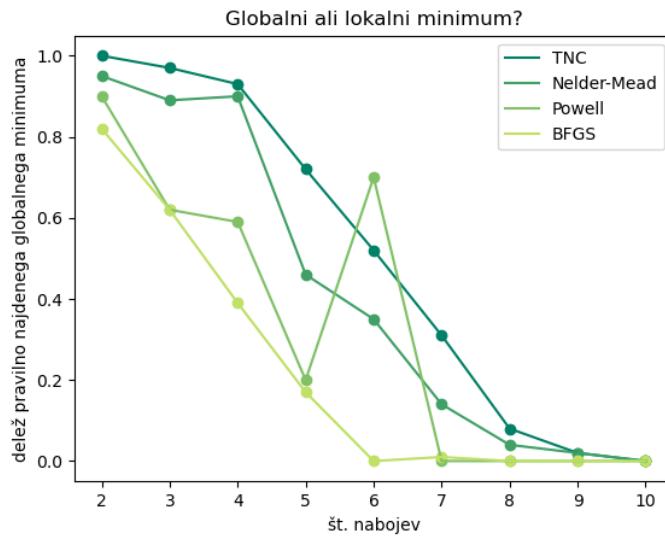


Slika 8: Program smo pognali nekajkat z naključnimi robnimi pogoji in dobili različne razporeditve. Nad vsako razporeditvijo je napisana energija, ki jo konfiguracija ima.

## 4 Krog

S tem, kar smo v prejšnjem poglavju ugotovili, v mislih lahko preverimo tudi kako se naboji razporedijo na polnem krogu. Kot v prejšnjem poglavju se večinoma razporedijo po robu, vendar pa se za večja

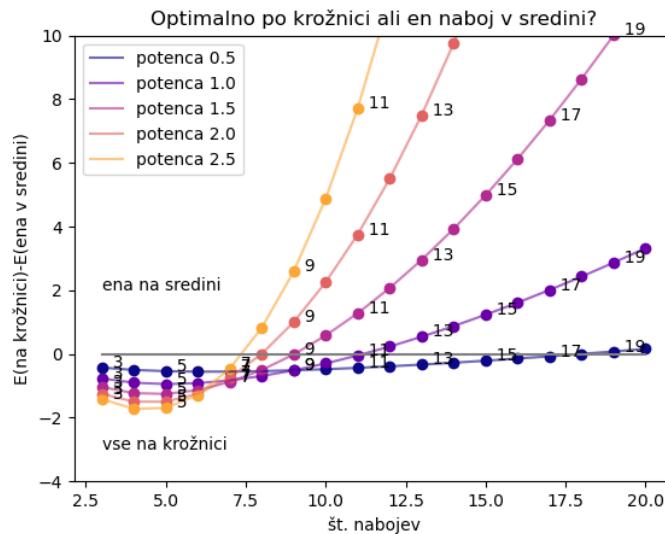
števila nabojev kakšen naboj včasih znajde tudi v sredini, glej sliko 8.



Slika 9: Prikaz deleža poskusov v katerih program najde globalni in v katerih najde lokalni minimum za različne vrednosti števila nabojev.

Kot vidimo je pri obeh (pri 5 in 9 nabojev) optimalna razporeditev na robu. Vendar se pojavljajo tudi lokalni minimumi z enim (ali celo dvema) nabojem znotraj kroga. Globalni minimum zelo redko najde, sploh pri več nabojev (za 9 nabojev sem morala program zagnati vsaj petkrat, da sem sploh dobila to razporeditev na enem od 16 grafov!). Poglejmo kolikšen je delež poskusov v katerih program najde globalni in ne lokalni minimum (slika 9).

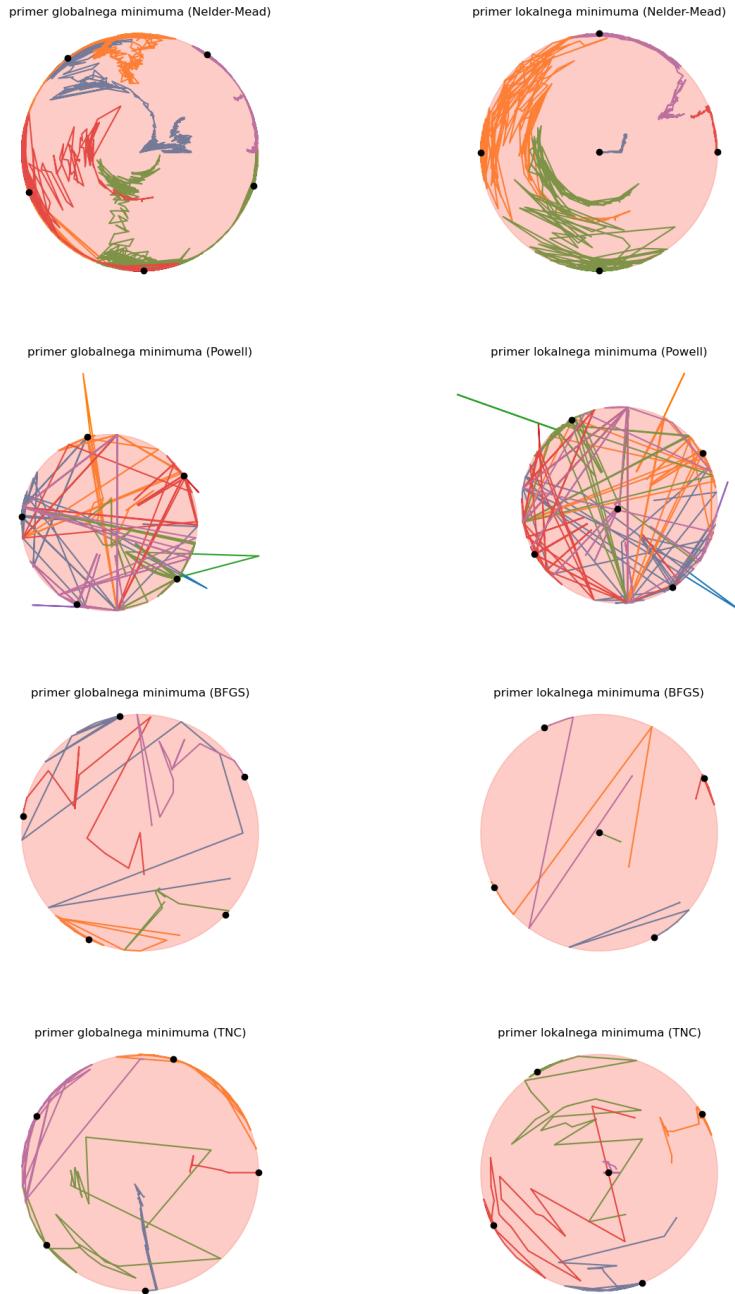
Vidimo, da ko gremo proti več nabojem, je delež pravilno najdenega globalnega minimuma vedno manjši. Vse pogosteje najde lokalni minimum. Vidimo lahko tudi primerjavo med metodami, najuspešnejši sta TNC in Nelder-Mead. Tudi tukaj velja načelo ”počasi se daleč pride”, saj sta obe ti dve metodi potrebovali bistveno več časa kot ostali dve.



Slika 10: Graf, ki nam kaže ali je bolj optimalna razporeditev na robu ali taka, da je eden izmed nabojev v sredini.

Sklepamo, da bo kakšen naboj, kljub preferirani legi na robu, ob nekem številu nabojev skočil na

sredino. Slika 10 prikazuje razliko med potencialom razporeditve nabojev po robu in razporeditve, kjer je en izmed nabojev v sredini. Kot pri torusu vidimo, da tudi tukaj pride v igro potenca s katero pada potencial. Kljub vsemu pa je zanimivo, da je pri elektrostatskem potencialu pri 11 nabojevih še vedno optimalna lega, da so vsi naboji na robu. Pri potencialu  $1/\sqrt{r}$  pa je ta meja še mnogo višje. Zanimivo je tudi, da graf 10 najprej pada (in to za vse potence), kar pomeni, da je bolj optimalno, če damo tri naboje v vrsto kakor pa če damo tri naboje v trikotnik in enega v središče.



Slika 11: Prikaz korakov po katerih različni algoritmi iščejo rešitev. Prikazana sta primera, ko algoritem najde lokalni minimum ter ko najde globalnega.

Krog pa je poleg zanimivih značilnosti, ki smo jih opisali zgoraj zelo uporaben tudi za prikaz delovanja zgoraj naštetih algoritmov. Na sliki 11 so prikazani koraki po katerih algoritmi iščejo minimum.

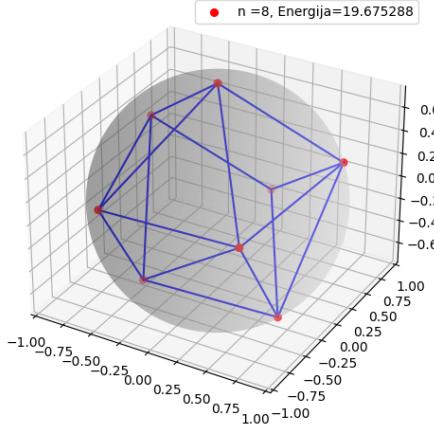
UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO  
ODDELEK ZA FIZIKO

MODELSKA ANALIZA I

**3. naloga: Numerična minimizacija**

Žiga Šinigoj, 28222025

Ljubljana, oktober 2022



Slika 3: Razporeditev naboja na enotski sferi pri osmih nabojih.

### 3 Dipoli na sferni lupini

(Nisem prepričan, da je to prava rešitev)

Zanimalo me je kako bi se razporedili dipoli po sferi oziroma sferni lupini, ki ima neko končno debelino in je izolirana od okolice. Polje, ki ga ustvarja dipol je

$$\mathbf{E}(\mathbf{r}) = \frac{3(\mathbf{p} \cdot \hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{p}}{4\pi\epsilon_0 r^3}. \quad (4)$$

Energijo med dvema električnima dipoloma lahko zapišem kot

$$E_{ij} = -\mathbf{p}_i \mathbf{E}_j = \frac{\mathbf{p}_i \mathbf{p}_j - 3(\mathbf{p}_i \hat{\mathbf{r}}_{ij})(\mathbf{p}_j \hat{\mathbf{r}}_{ij})}{4\pi\epsilon_0 r_{ij}^3}, \quad (5)$$

kjer je  $\hat{\mathbf{r}}_{ij} = \frac{\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j}{|\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j|}$ ,  $\mathbf{p}_i$  dipolni moment,  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|^3$ . Energija je sedaj odvisna od razdalje med dipoloma in tudi njihove orientacije. Skupno energijo sistema N dipolov lahko zapišem kot

$$E = \sum_{i < j} E_{ij} = \sum_{i < j} -\mathbf{p}_i \mathbf{E}_j = \sum_{i < j} \frac{\mathbf{p}_i \mathbf{p}_j - 3(\mathbf{p}_i \hat{\mathbf{r}}_{ij})(\mathbf{p}_j \hat{\mathbf{r}}_{ij})}{4\pi\epsilon_0 r_{ij}^3}. \quad (6)$$

Enako kot v primeru nabojev lahko zapišem energijo v brezdimenzijski obliki kot

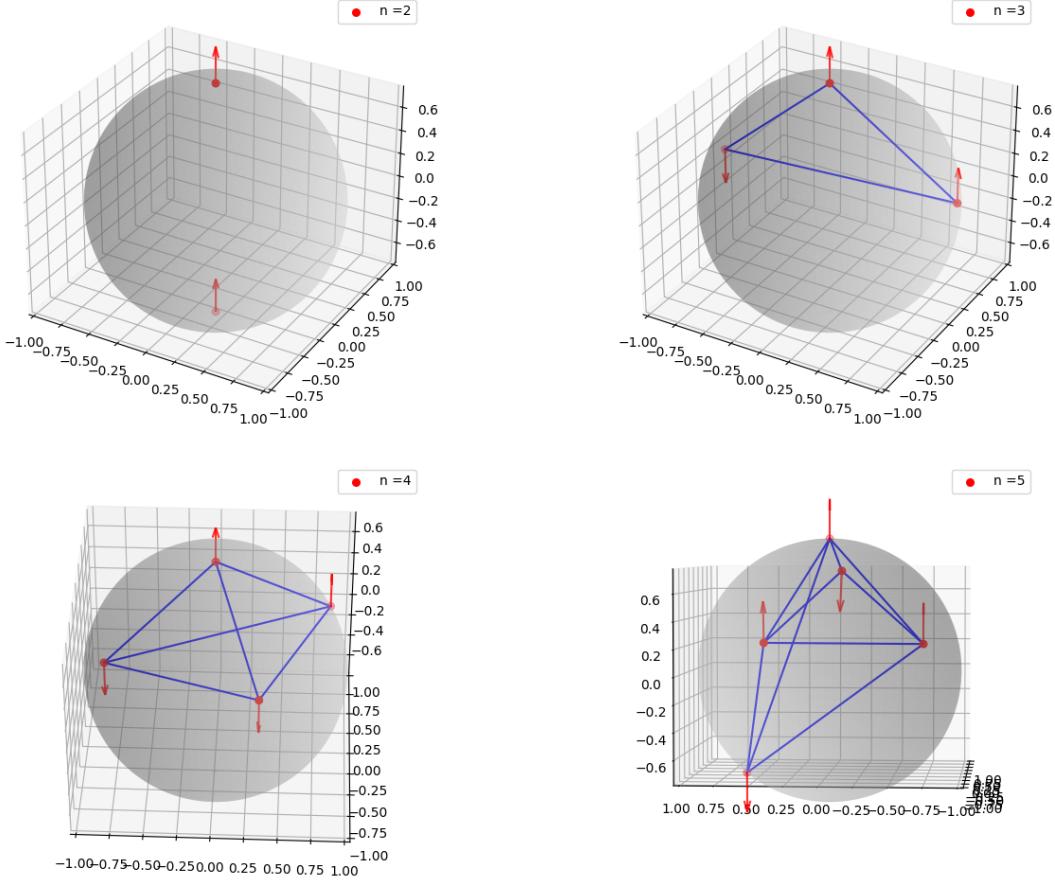
$$E = \sum_{i < j} \frac{\tilde{e}_i \tilde{e}_j \hat{\mathbf{d}}_i \hat{\mathbf{d}}_j - 3\tilde{e}_i \tilde{e}_j (\hat{\mathbf{d}}_i \hat{\mathbf{r}}_{ij})(\hat{\mathbf{d}}_j \hat{\mathbf{r}}_{ij})}{r_{ij}^3}, \quad (7)$$

kjer sem vzel direktorje  $\mathbf{d}_i$  za enotske. Sedaj ima vsak dipol 4 parametre. Kota  $\vartheta$ ,  $\varphi$ , ki določata lokacijo dipola na enotski sferi in kota  $\vartheta_d$ ,  $\varphi_d$ , ki določata orientacijo dipola. Prvi dipol lahko postavim na  $\mathbf{r}_0 = (0, 0, 1)$ ,  $\mathbf{d}_0 = (0, 0, 1)$ . Energija je sedaj funkcija 4(N-1) parametrov za N dipolov, kar privede na reševanje

$$E(\varphi_1, \vartheta_1, \varphi_{1d}, \vartheta_{1d}, \dots, \varphi_{N-1}, \vartheta_{N-1}, \varphi_{(N-1)d}, \vartheta_{(N-1)d}) = \min \quad . \quad (8)$$

Pri minimizaciji energije sem imel kar nekaj težav, saj je edino metoda Nelder-Mead končala uspešno minimizacijo, ampak je dala velike energije ( $> \pm 10^3$ ). Ostale metode na primer BFGS so končale neuspešno, saj naj nebi uspele zagotoviti dane natančnosti, kjub temu, da sem jo zmanjšal. Vseeno sem se po ogledu rezultatov in energij, ki so pri neuspešnih metodah bile manj kot  $10^{-3}$ , določil za slednje rešitve. Pomembna je bila tudi izbira začetnih parametrov, saj sem pri naključni izbiri dobil slabe minimizacije. V ta namen sem vsak začetni dipol nasprotno obrnil ( $\hat{\mathbf{d}}_{i0} = (-1)^i \hat{\mathbf{d}}_{00}$ ), kar je veliko pripomoglo k minimizaciji. Konfiguracije dipolov za različna števila so prikazane na sliki 4.

Konfiguracija za 2 dipola je smiselna, saj na ta način dosežemo najmanjšo energijo. Prav tako imajo ostale nasprotne zasukane dipole in tvorijo deformirane like iz rešitev za monopole (4 in 5 dipolov), kar se mi zdi tudi fizikalno smiselno.



Slika 4: Razporeditev dipolov na enotski sferi pri različnem številu dipolov.

## 4 Semafor

Nalogo o semaforju iz prve vaje modelske analize lahko rešim tudi na diskreten način. Reševal bom primer, ko imamo eno krajišče gibljivo. Za minimizacijo sem uporabil algoritem BFGS. Interval razdelim na  $N$  ekvidistančnih točk  $\tilde{t}_0, \dots, \tilde{t}_{N-1}$ , ki so razmaknjene za  $\delta\tilde{t}$ . Na vsakem intervalu imam hitrost  $\tilde{v}_i = \tilde{v}(\tilde{t}_i)$ . Želim minimizirati funkcional

$$F = \int_0^1 \dot{\tilde{v}}^2(\tilde{t}) d\tilde{t} = \min ,$$

ki ga prepišem v diskretno obliko z uporabo trapezne formule in končnih differenc.

$$F = \left( \left( \frac{\tilde{v}_1 - \tilde{v}_0}{\delta\tilde{t}} \right)^2 + \dots + \frac{1}{2} \left( \frac{\tilde{v}_{N-1} - \tilde{v}_{N-2}}{\delta\tilde{t}} \right)^2 \right) \delta\tilde{t}. \quad (9)$$

Vez

$$Z = \int_0^1 \tilde{v}(\tilde{t}) d\tilde{t} = 1 \quad (10)$$

lahko upoštevam na dva načina.

### 4.1 Eksaktna vez

Vez lahko diskretiziram

$$Z = \left( \frac{1}{2}\tilde{v}_0 + \tilde{v}_1 + \dots + \frac{1}{2}\tilde{v}_{N-1} \right) \delta\tilde{t} \quad (11)$$

in sedaj iščem minimum funkcionala

$$\tilde{F} = F - \lambda Z = \min . \quad (12)$$

## 2. Naloga

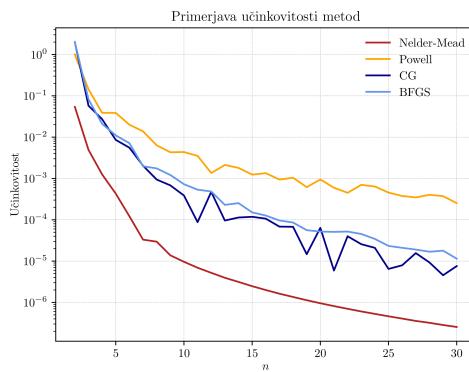
- Partnerjavnost z celotnim podzemjem
- Odvisnost od dobre krovije
- Reslovne izmedse mest / letogramična
- Pot.

# Dance kacvjanot

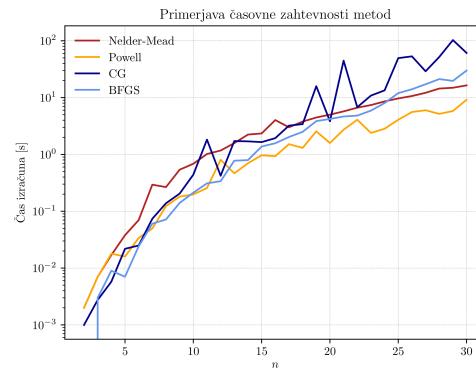
Poskušajmo ovrednotiti uporabljeni metode. Rezultati minimizacije se za posamezne metode razlikujejo šele na deseti decimalki, zato poiščimo kak drug kriterij. Definiramo učinkovitost metode kot:

$$\text{učinkovitost} \equiv \frac{1}{\text{število iteracij} \times \text{rezultat}}, \quad (1)$$

kjer je rezultat končna vrednost minimizirane funkcije. Manjše kot je število iteracij in bolj kot je funkcija minimizirana, večja je torej učinkovitost. Iz slike 4 lahko sklepamo, da je glede na izbran kriterij najmanj učinkovita metoda Nelder-Mead, najbolj pa Powellov algoritem. Na sliki 5 je prikazana časovna zahtevnost posamezne metode. Tudi po tem kriteriju je metoda Powell najboljša izbira, za izračun rešitve pa največ časa porabi metoda konjugiranih gradientov. Na tem mestu je vredno omeniti tudi dejstvo, da metoda porabi več časa, če naredi več iteracij. Kriterija učinkovitosti in hitrosti izračuna, zato nista popolnoma neodvisna. Zanimivo je tudi dejstvo, da je ena od privzetih metod funkcije `scipy.optimize.minimize` BFGS algoritem, od katerega je po naših izbranih kriterijih boljša Powellova metoda.



Slika 4: Učinkovitost posameznih metod glede na kriterij (1).



Slika 5: Časovna zahtevnost posameznih metod.

## Semafor

Ponovno rešujemo problem semforjev. Naš cilj je prevoziti semafor v trenutku, ko se prižge zelena luč. Lagrangian je enak kot v 1. nalogi, torej:

$$\mathcal{L} = \int \left( \frac{dv}{dt} \right)^2 dt - \lambda \int v dt = \min.$$

Lagrangian diskretiziramo in prepišemo v bredimensijsko obliko. Minimizacijo pospeška zahtevamo s členom:

$$F_0 = \frac{1}{2} \frac{y_1^2}{\Delta \tau} + \sum_{i=2}^{N-1} \frac{(y_i - y_{i-1})^2}{\Delta \tau} + \frac{1}{2} \frac{(y_N - y_{N-1})^2}{\Delta \tau},$$

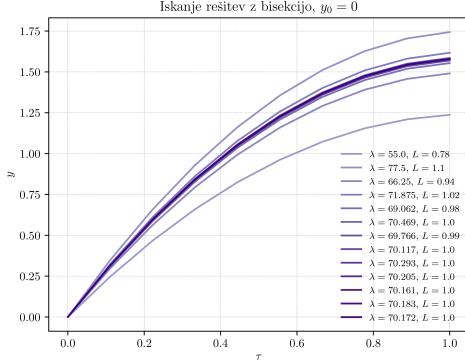
pogoju o prevoženi poti pa zadostimo z

$$F_1 = \frac{1}{2} y_1 \Delta \tau + \sum_{i=2}^{N-1} y_i \Delta \tau + \frac{1}{2} y_N \Delta \tau.$$

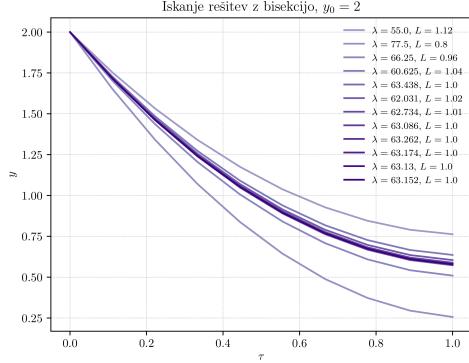
Funkcija, ki jo minimiziramo je

$$F = F_0 - \Lambda F_1.$$

Funkcijo minimiziramo za izbrane vrednosti  $\Lambda$ , nato pa s pomočjo bisekcije poiščemo takšen  $\Lambda$ , da zadostimo pogoju o prevoženi poti. Iskanje rešitev za dve različni začetni hitrosti s pomočjo bisekcije je prikazano na slikah 6 in 7.



Slika 6: Iskanje rešitev z bisekcijo za začetno hitrost  $y_0 = 0$ . Začetni vrednosti  $\Lambda$  sta bili 55 in 75.



Slika 7: Iskanje rešitev z bisekcijo za začetno hitrost  $y_0 = 2$ . Začetni vrednosti  $\Lambda$  sta bili 55 in 70.

Lagrangianu lahko dodamo tudi kakšno omejitev glede hitrosti. Pri 1. nalogi so nam na primer delale težave negativne vrednosti hitrosti. To nenegativnost hitrosti lahko zagotovimo s členom

$$F_2 = \sum_{i=1}^N e^{-\beta y_i}.$$

Večji kot je parameter  $\beta$ , močnejša je omejitev hitrosti na pozitivne vrednosti. Biti moramo pazljivi, da ta člen ne prevlada nad pogojem o prevoženi poti. Funkcija, ki jo minimiziramo je zdaj

$$F = F_0 + \Lambda F_1 + F_2,$$

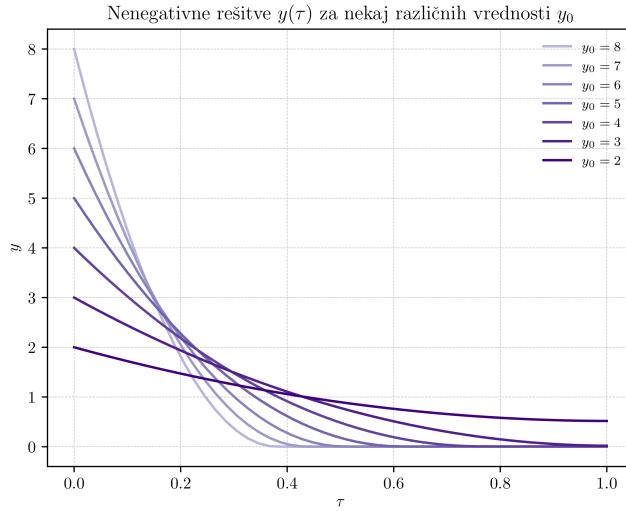
rešitve pa so prikazane na sliki 8. Vidimo, da tudi za zelo velike začetne hitrosti vrednosti niso negativne. Rešitve, ki smo jih dobili, pomenijo, da se pri semaforju ustavimo, saj tam vrednost hitrosti doseže ničlo.

V zadnjem koraku poskušajmo dobiti rešitve za zaporedne semaforje, ki so drug od drugega oddaljeni za različne dolžine, prav tako pa se razlikujejo časi do prižiga zelene luči. Število semaforjev, ki jih želimo prevoziti je  $M$ , dolžini poti in časovnega intervala od enega do drugega semaforja pa sta  $T_j$  in  $L_j$ . S členom  $F_{\text{zaporedni}}$  zagotovimo, da v času  $T_j$  prevozimo razdaljo  $L_j$ .

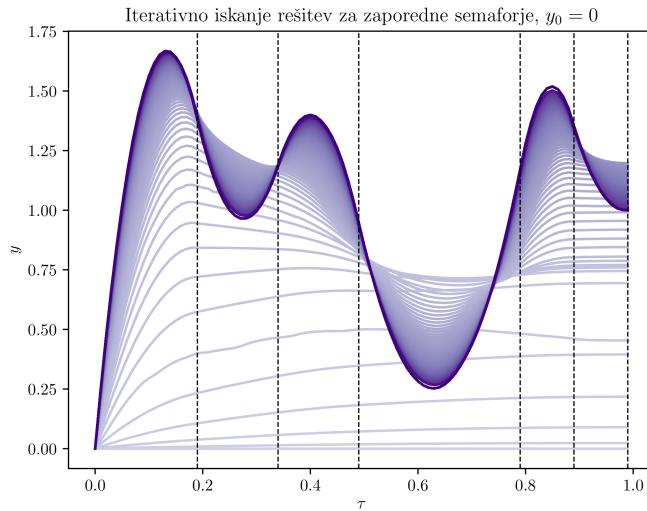
$$F_{\text{zaporedni}} = \sum_{j=1}^M \exp \left[ \pm \kappa \left( \frac{y_1}{2} + y_2 + \cdots + y_{N_j-1} + \frac{y_{N_j}}{2} - \frac{L_j}{\Delta \tau} \right) \right].$$

Večji kot je parameter  $\kappa$ , bolj je izpolnjena zahteva. Če je vstopna hitrost  $y_1 < \frac{L_j}{T_j}$ , torej manjša od hitrosti, s katero bi enakomerno prevozili interval, je predznak minus. Če velja  $y_1 > \frac{L_j}{T_j}$  pa plus. S tem zagotovimo, da pospešujemo, če je hitrost premajhna in vozimo pojemajoče, če je ta prevelika. Končno lahko zapišemo funkcijo, ki jo želimo minimizirati.

$$F = F_0 + F_{\text{zaporedni}}$$

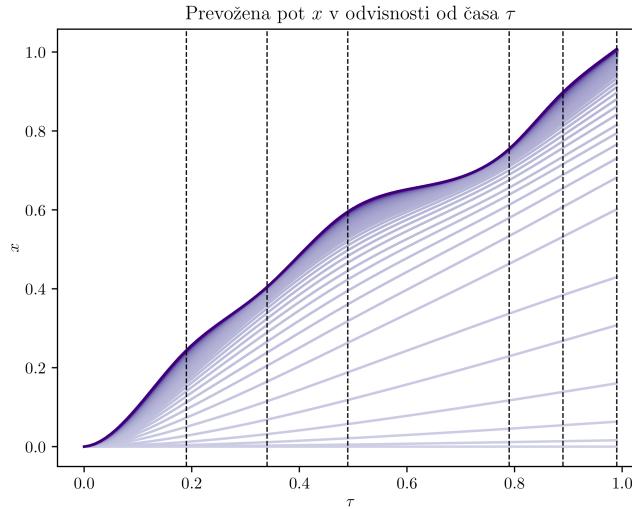


Slika 8: Nenegativne rešitve  $y(\tau)$  za velike vrednosti  $y_0$ .

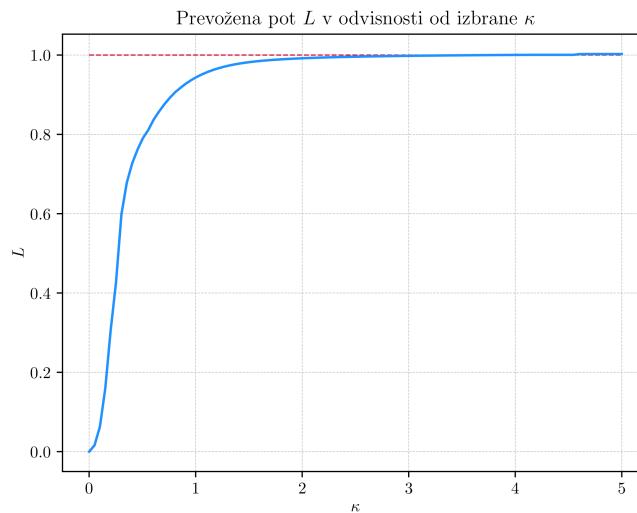


Slika 9: Rešitve za zaporedne semaforje  $y(\tau)$  za različne vrednosti parametra  $\kappa$ .

Problem rešimo iterativno. Začetna vrednost  $\kappa$  je 0, kar nam da nepospešeno gibanje, saj minimiziramo le pospešek. Rešitev iz prejšnjega koraka uporabimo za približek v naslednjem in parameter  $\kappa$  postopoma povečujemo. Proses iskanja rešitev in končno rešitev prikazuje slika 9. Pot  $x(\tau)$  pa prikazuje slika 10. Slika 11 prikazuje konvergenco dolžine poti k pravi vrednosti v odvisnosti od parametra  $\kappa$ .



Slika 10: Pot  $x(\tau)$  za zaporedne semaforje za različne vrednosti parametra  $\kappa$ .



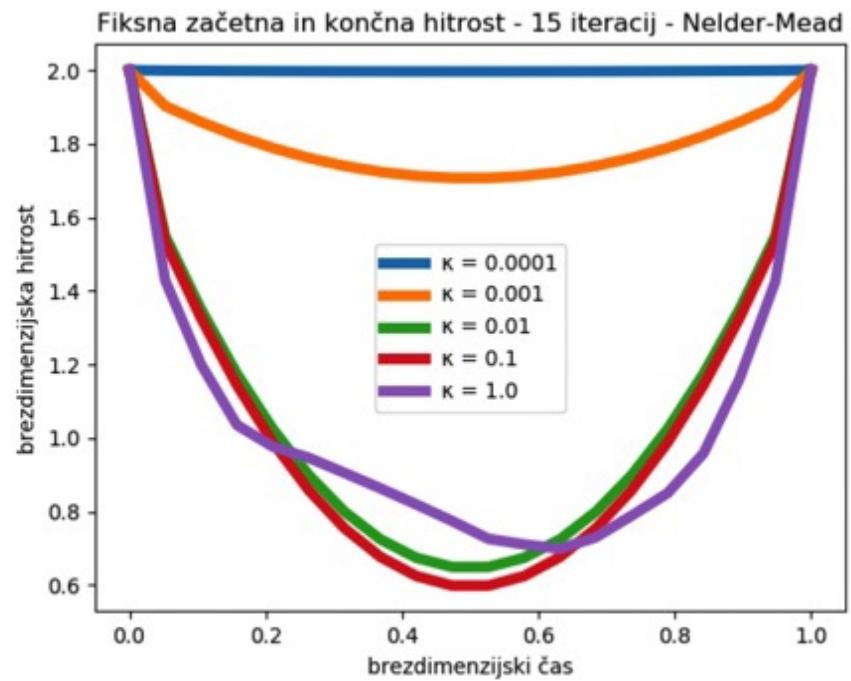
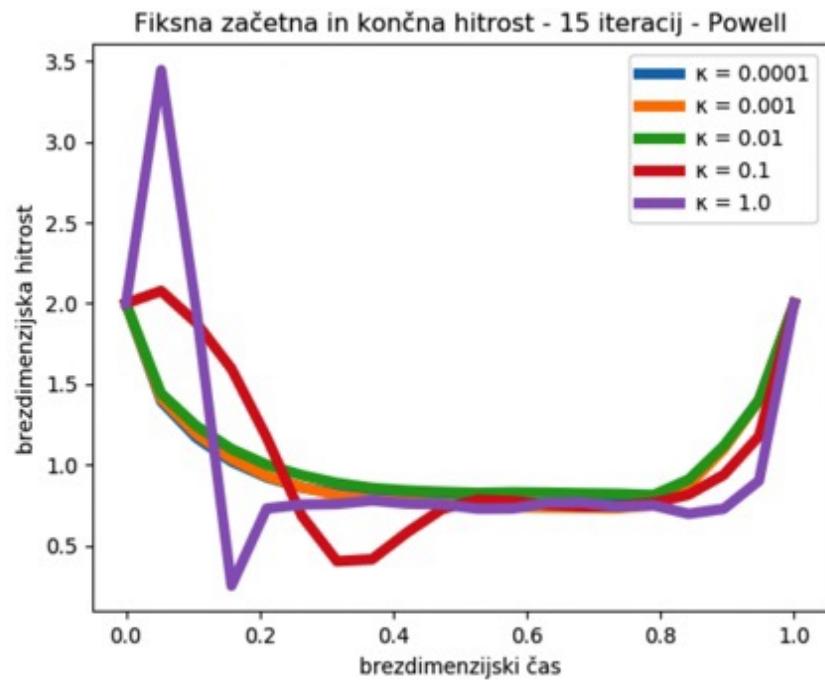
Slika 11: Skupna prevožena pot  $L$  v odvisnosti od parametra  $\kappa$ .

## Zaključek

V nalogi smo se spoznali z metodo numerične minimizacije in jo implementirali na Thomsonov problem in problem optimalne vožnje skozi semafor. Primerjali smo več metod in si pogledali, kako se premakne težišče, če v pol postavimo naboj, ki je nekajkrat večji od ostalih. V drugem delu naloge smo uspeli optimizirati tudi vožnjo skozi več zaporednih semaforjev.

*Ponmerjova s teorijo.*

?



Slika 4: Grafični prikaz rešitev za Powellovo in Nelder-Meadovo metodo.

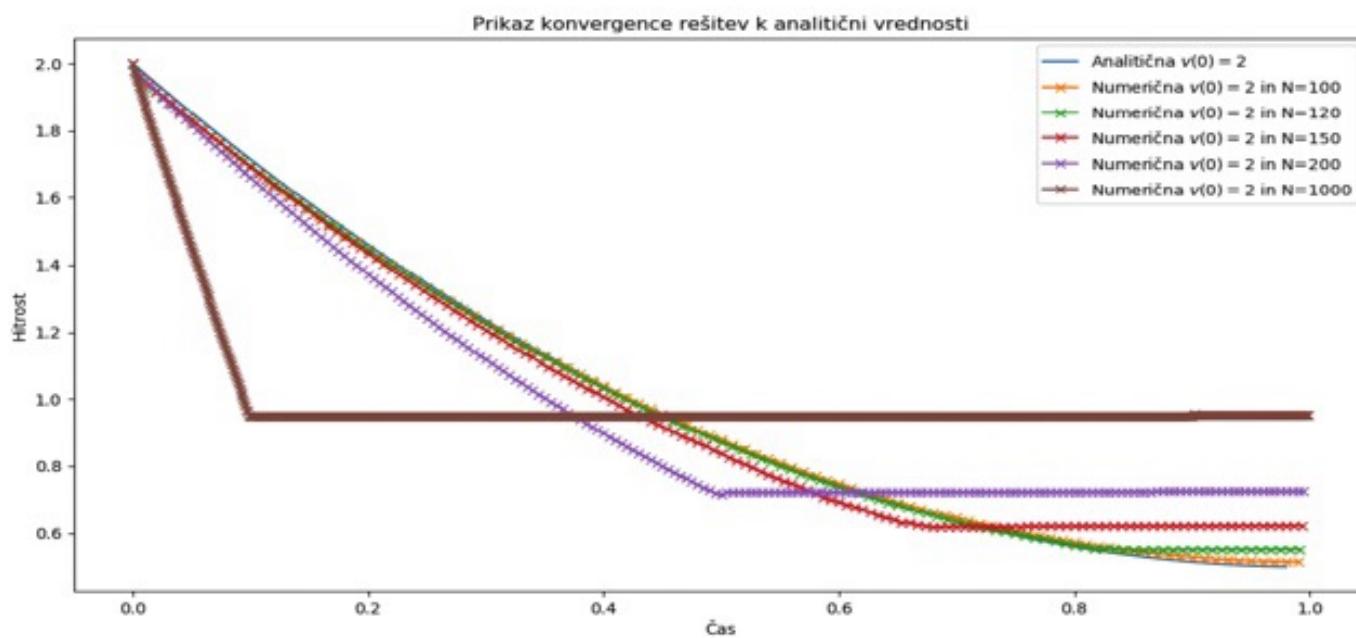
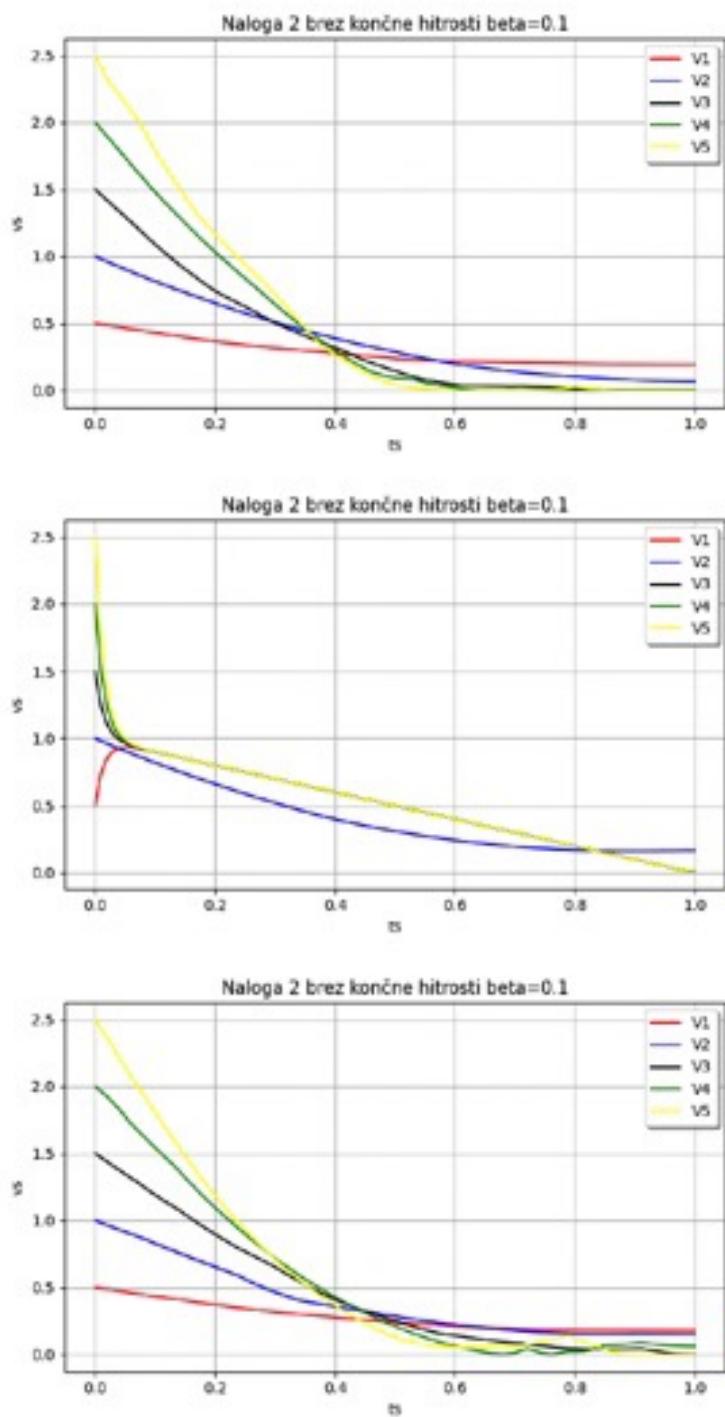


Figure 27: Prikaz konvergencije krivulje prostega končnega parametra za pretirane velikosti  $N$



Slika 4: Prikaz rezultatov za različno začetno hitrost in poljubno končno hitrost ob različni izbiri začetnega ugibanja.

Dodecah:

Na sliki 9 vidim, da so rešitve za velik  $\kappa$  zelo dober približek analitični, saj relativne napake praktično ni. Na začetku sem mislila, da sem se kje v postopku zmotila, ampak ker je za vse priemre koda napisana na enak način in ker ne dobim relativne napake enake 0 za vse primere, bo že moralo držati. Za  $\kappa = 0.1$  vidim, da rešitve hitro pospešijo, nato pa se gibanje nadaljuje enakomerno. Na desni strani, ko sem spremajala parameter  $\kappa$ , sem y skalo omejila na [-2.5, 5.5], da se vidi ujemanje z analitično rešitvijo pri velikih  $\kappa$ . Odstopanja rešitev izven tega okvira pa se vidi na relativnih napakah in na desnem grafu pri  $\kappa = 0.1$ . Rešitve so ali zelo dobre, ali pa zelo slabe in se hitro menjajo glede na vrednost parametra  $\kappa$ , kar pa je nekako v naravi z eksponentno funkcijo.

## 2.3 Jedilnik

Ker sem v nalogi 103 že obiskala nalogo 101, bom obiskala še nalogo 102 in minimizirala kalorije z pomočjo funkcije `minimize` modula `scipy.optimize`.

Vzela sem eksaktne vezi, kot so prikazane v tabeli 1 minimizirala pa sem kalorije. Po postopku iz naloge 102 dobila rezultat, prikazan na sliki 10.

hranolna snov	dnevni vnos
mašcobe	= 70 g
ogljikovi hidrati	= 310 g
protein	= 50 g
kalcij	= 1000 mg
železo	= 18 mg

Tabela 1: Tabela dnevnih vnosov hranilnih snovi.

Na začetku sem se postopka lotila preko Lagrangevih multiplikatorjev. Želela sem si minimizirati kalorije, torej

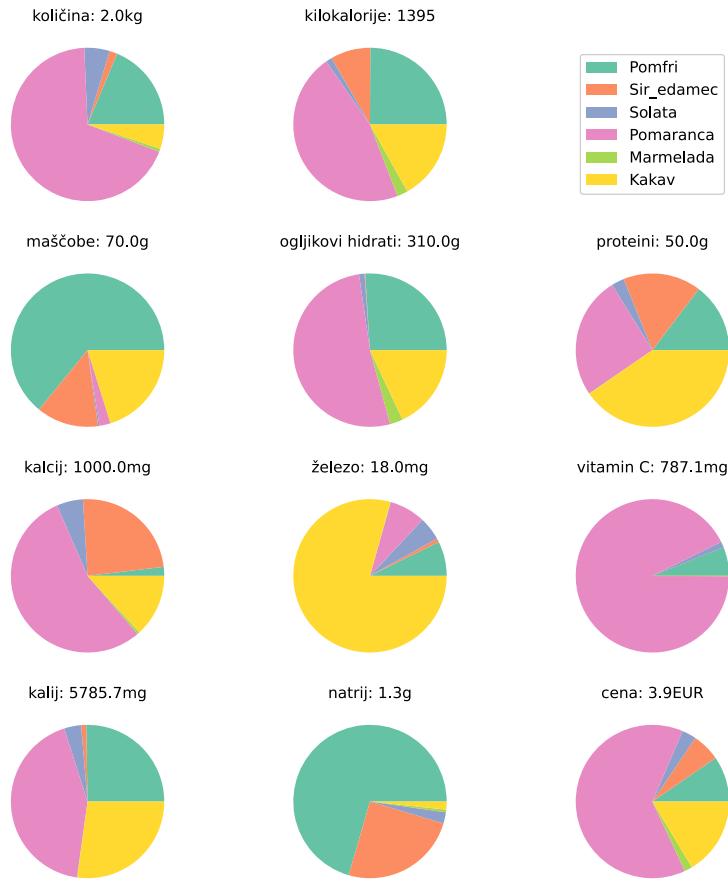
$$F = \mathbf{x} \cdot \mathbf{a}_{\text{kal}}.$$

Funkcional  $\tilde{F}$  pa zaradi vezi vsebuje 6 multiplikatorjev

$$\tilde{F} = F - \sum_{i=1}^6 \lambda_i \mathbf{x} \cdot \mathbf{a}_i,$$

kjer so  $\mathbf{a}_i$  vektorji hranilnih snovi (npr.  $\mathbf{a}_1$  vsebuje vsebino mašcob). Po podobnem postopku kot pri optimalni vožnji v poglavju 2.2.1, sem minimizirala funkcional  $\tilde{F}$  za nek vektor  $\lambda$  z komponentami  $\lambda_i$ . Ko sem dobila  $\mathbf{x}(\lambda)$ , sem definirala funkcijo, ki je sprejela vektor  $\lambda$  in vrnila vektor s komponentami  $\mathbf{x} \cdot \mathbf{a}_i - b_i$ , kjer je  $b_i$  vrednost hranilne snovi (npr. 70 pri mašcobah). Uporabljala sem metodo L-BFGS-B, saj ta sprejema tudi meje (v tem primeru nobene hrane ne more biti manj kot nič). Na žalost pa se je izkazalo, da po tem načinu ne gre. Vrednosti so zelo divergirale. Na kakeršenkoli način sem poskusila numerično zapisati količine, sem imela isti problem - izbira začetnega približka za vektor

$\lambda$ , večinokrat se od začetnega približka niti ni premaknil, velikokrat pa je javil, da se je minimizacija izvedla brez napak, a so bile vse vrednosti nič ali pa zelo velike. Ure sem poskušala spremenjati pogoje, da bi metoda našla minimum, a se na žalost ni izšlo. Poskušala sem tudi zmanjšati število Lagrangevih multiplikatorjev in s tem število vezi - tudi s tem nisem dosegla nič.



Slika 10: Tortni diagrami količin, kilokalorij, cene ter hranilnih snovi. Pridobljen po načinu reševanja, kot je v nalogi 102.

Potem sem poskusila še preko reševanja z eksponentnimi členi. Zapisla sem funkcional

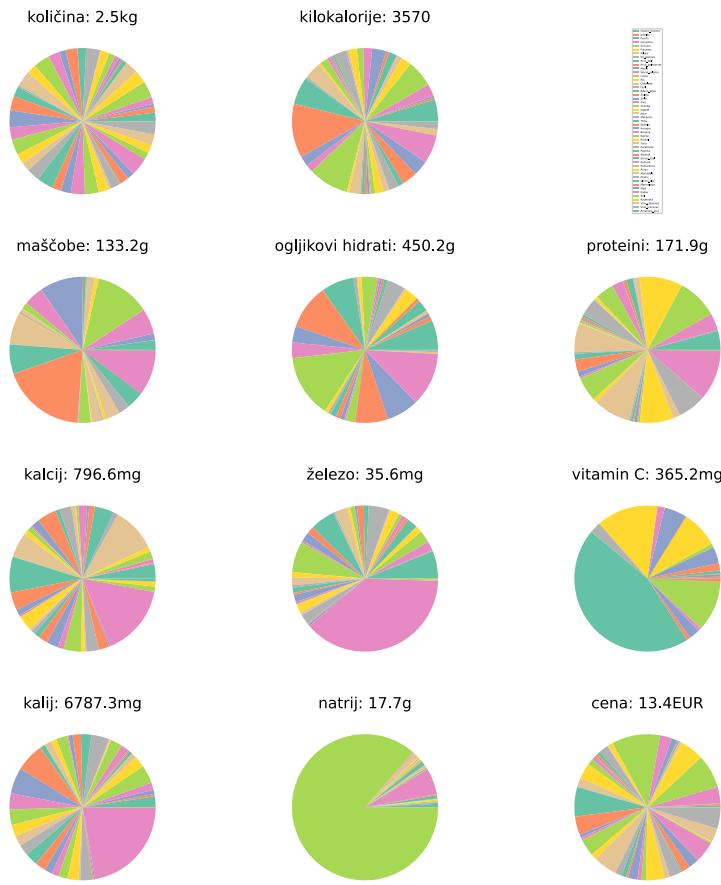
$$\tilde{F} = F + \sum_{i=1}^6 e^{\kappa(\mathbf{x} \cdot \mathbf{a}_i - b_i)},$$

ki sem ga poskusila minimizirati pri nekem parametru  $\kappa$ . Na začetku sem si izbrala  $\kappa = 1$  - ni šlo, potem  $\kappa = 10$  - ni šlo, tudi če sem za začetni približek nastavila rešitev iz linearnega programiranja - ni šlo. Iz jeze sem vtipkala  $\kappa = 100000000000$  - in je uspelo. Dobila sem enako rešitev, kot preko linearnega programiranja (slika 10). Vendar samo za zelo previdno izbrane začetne približke. Če si začetni približek izbrem preko funkcije `random`, dobim rezultat prikazan na sliki 11, kjer je vsake hrane po malo, vidi pa se tudi da vezi niso upoštevane. Globalnega minimuma v tem primeru ne najde.

Poskusila sem tudi definirati funkcional kot

$$\tilde{F} = F + \prod_{i=1}^6 e^{\kappa(\mathbf{x} \cdot \mathbf{a}_i - b_i)},$$

a sem imela še hujši problem - niti z skrbno izbranim približkom ni našel globalnega minimuma.



Slika 11: Tortni diagrami količin, kilokalorij, cene ter hranilnih snovi. Pridobljen po načinu reševanja, kot je v (tej) nalogi 103.