



UNIVERZA  
V LJUBLJANI

FMF

Fakulteta za matematiko  
in fiziko

---

# Nelinearno programiranje

---

*Modelska analiza, vaja 3*

**Avtor:** Urban Pečoler  
**Vpisna številka:** 28242019

Oktober 2024

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Reševanje problema</b>	<b>1</b>
2.1	Energija stabilnih ravnovesnih stanj . . . . .	2
2.2	Volumen poliedra . . . . .	4
<b>3</b>	<b>Analiza različnih algoritmov minimizacije</b>	<b>5</b>
3.1	Uspešnost minimizacije . . . . .	6
3.2	Dipolni moment . . . . .	6
<b>4</b>	<b>Dodatno: Geometrija 2D kroga</b>	<b>7</b>
<b>5</b>	<b>Optimalna vožnja skozi semafor</b>	<b>9</b>
<b>A</b>	<b>Thomsonov problem za različne N</b>	<b>12</b>

## 1 Uvod

Cilj Thomsonovega problema je določiti konfiguracijo z minimalno elektrostatično potencialno energijo za  $N$  elektronov, ki so omejeni na površini enotne sfere ( $r = 1$  in se med seboj odbijajo z močjo, določeno z Coulombovim zakonom). Naboji se poradelijo tako, da je v dani porazdelitvi najmanjša možna elektrostatska energija:

$$E = \sum_{i,j,i \neq j} \frac{e_0}{4\pi\epsilon_0 |\mathbf{r}_i - \mathbf{r}_j|} = \min \quad (1)$$

Zaradi lažjega računanja bom predpostavil, da je  $e_0 = \epsilon = 1$ , torej v ustreznih enotah velja:

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

Ker imamo geometrijo krogle, se premaknemo tudi v sferične koordinate. Položaj  $i$ -tega naboja na krogli je tako

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

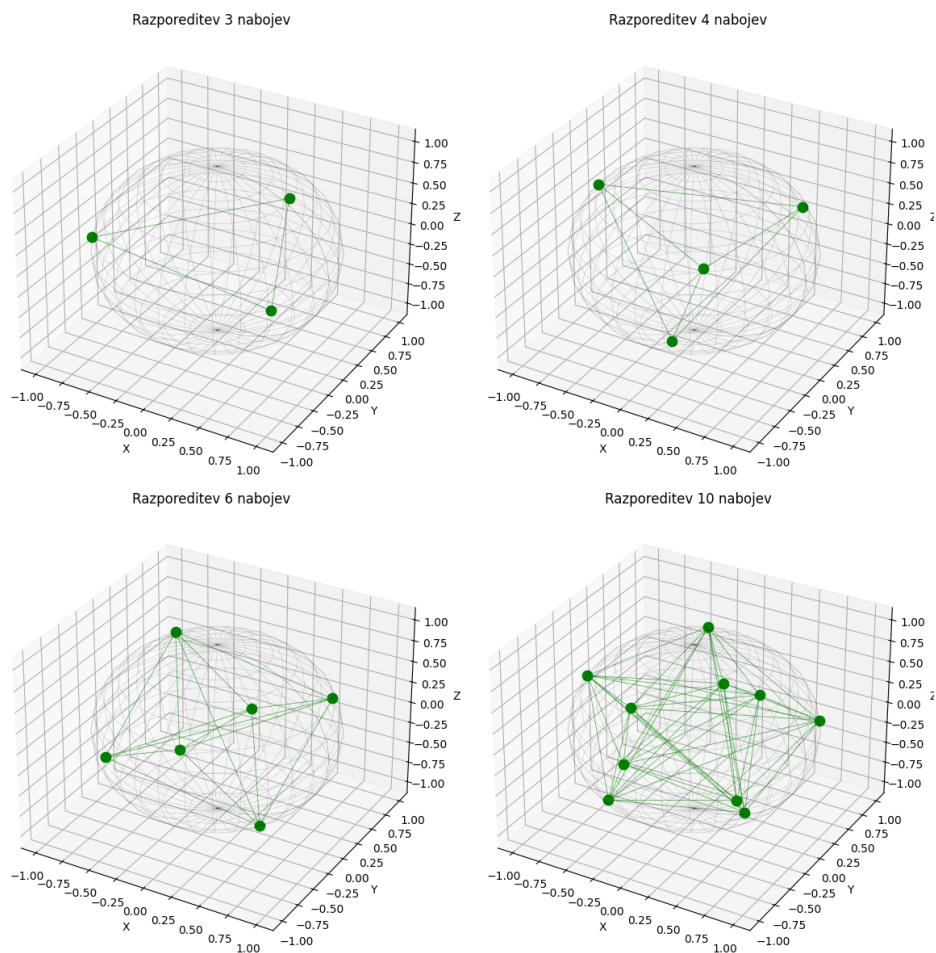
kjer s  $\theta_i$  in  $\varphi_i$  označimo polarni in azimutalni kot delca. Minimiziramo torej funkcijo

$$E(\mathbf{X}) = \sum_{i < j} \frac{1}{\sqrt{1 - \sin \theta_i \sin \theta_j \cos(\varphi_i - \varphi_j) - \cos \theta_i \cos \theta_j}} \quad (3)$$

kjer je  $\mathbf{X} = (\theta_1, \varphi_1, \theta_2, \varphi_2, \dots, \theta_N, \varphi_N)$ .

## 2 Reševanje problema

Problem sem reševal s pomočjo programskega jezika *Python* in funkcije `scipy.optimize.minimize` iz knjižnice *scipy* [1]. Funkcija ima možnost izbire algoritma minimizacije, kar sem v nadaljni analizi s pridom izkoristil, da sem primerjal hitrosti in natančnosti algoritmov. Za začetek pa sem si samo pogledal razporeditev nabojev na sferi za  $N = 3, 4, 6, 10$  nabojev. Pri risanju grafov, sem si začetne pogoje izbral naključno, s pomočjo funkcije `numpy.random.randint` ter izbral minimizacijsko metodo "Nelder-Mead" oz. Ameba.

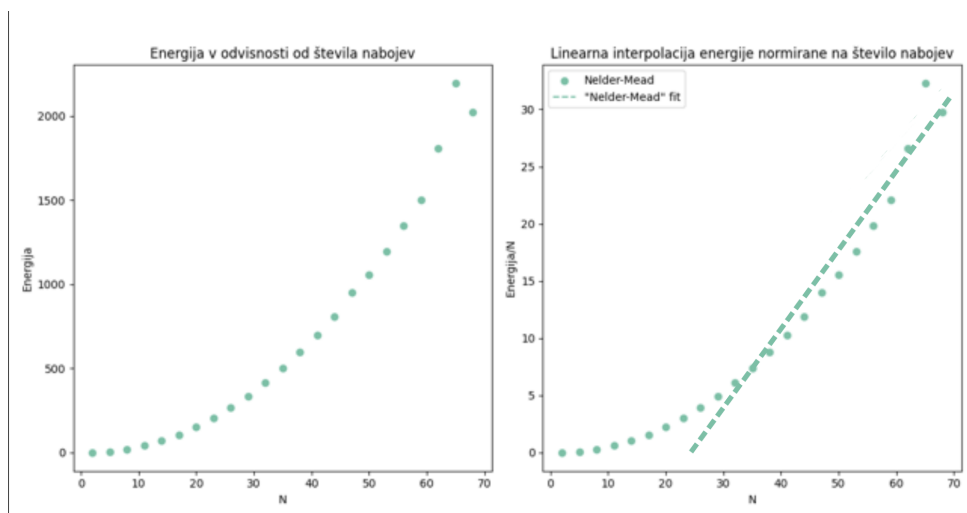


Slika 1: Rešitve Thomsonovega problema za različno število nabojev ( $N = 3, 4, 6, 10$ ) na krogli. Z zelenimi pikami narisani naboji, zelene tanke črte pa so stranice poliedra, ki ga točke objemajo. Sfera je narisana z tanko sivo obrobo. Za minimizacijo je uporabljen algoritem "Nelder-Mead".

Slika 1 prikazuje razporeditev nabojev pri minimalni energiji. Vidimo, da v stabilni ravnovesni legi, dobimo različne poliedre. Za 4 naboje je to tetraeder, za 6 bipiramida itd. Grafe za še več različnih nabojev pa sem dodal v dodatku A.

## 2.1 Energija stabilnih ravnovesnih stanj

Najbolj očitno vprašanje, ki se pojavi pri analizi tega problema je kakšna je odvisnost energije od števila nabojev na sferi. Z metodo "Nelder-mead" sem te energije izračunal, ter ugotovil, da ima energija, ki je normirana na število delcev na sferi linearno odvisnost od  $N$ . Linearna odvisnost se pojavi pri večjih  $N$ , pri  $N \leq 10$ , se funkcija ne prilega premici.

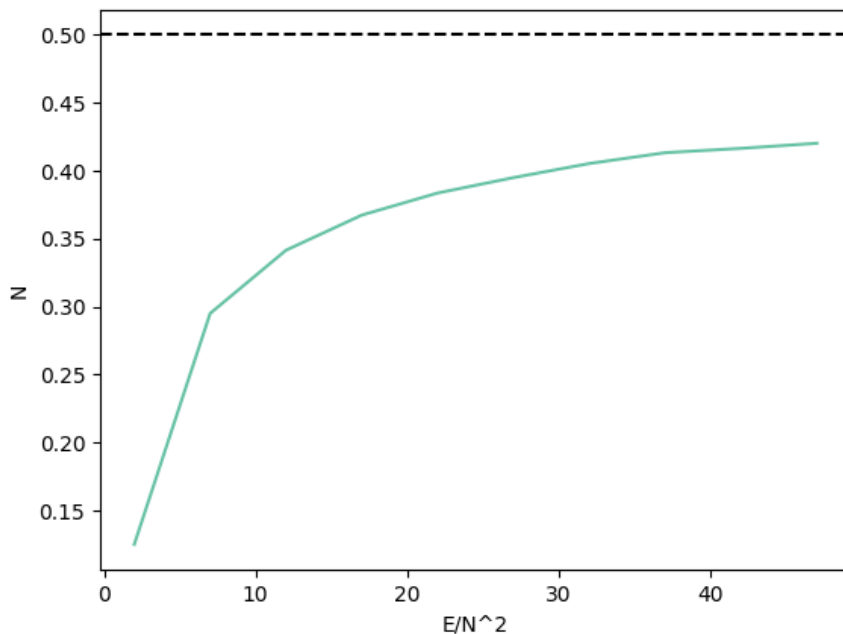


Slika 2: [LEVO]Energije rešitev Thomsonovega problema za različna števila nabojev na krogelni lupini, pridobljen z metodo "Nelder-Mead". Graf spominja na kvadratično funkcijo. [DESNO] Linearni fit za funkcijo  $E/N$ , za  $N > 15$ .

Zanimivo je tudi pogledati, kaj se dogaja z energijo, ko gre  $N \rightarrow \infty$ . Takrat se bližamo primeru zvezne ploskovne porazdelitve naboja. Po naši definiciji energije lahko pričakujemo, da

$$\lim_{N \rightarrow \infty} \frac{E}{N^2} = \frac{1}{2} \quad (4)$$

Na sliki 3 lahko to hipotezo potrdimo, čeprav je konvergenca dokaj počasna. Graf sem narisal le do  $N = 70$ , saj so algoritmi minimizacije za večje  $N$  zelo počasni.



Slika 3: Konvergenca  $E/N^2$  k zvezni porazdelitvi naboja. Rezultat je pridobljen preko metode "Nelder-Mead". Z črtkano črto je prikazana asimptota h kateri funkcija konvergira.

## 2.2 Volumen poliedra

Kot smo lahko videli na sliki 1, razporejeni naboji na krogli tvorijo različne poliedre z različnimi dolžinami stranic. Zanimalo me je, kako se povprečna dolžina stranic med sosednjima nabojema spreminja z  $N$  in kako se z  $N$  spreminja volumen poliedra. Za analizo tega problema sem uporabil funkcijo `scipy.spatial.ConvexHull` [2], ki omogoča izračun konveksne lupine točk, kar predstavlja polieder, ki ga tvorijo naboji na površini krogle.

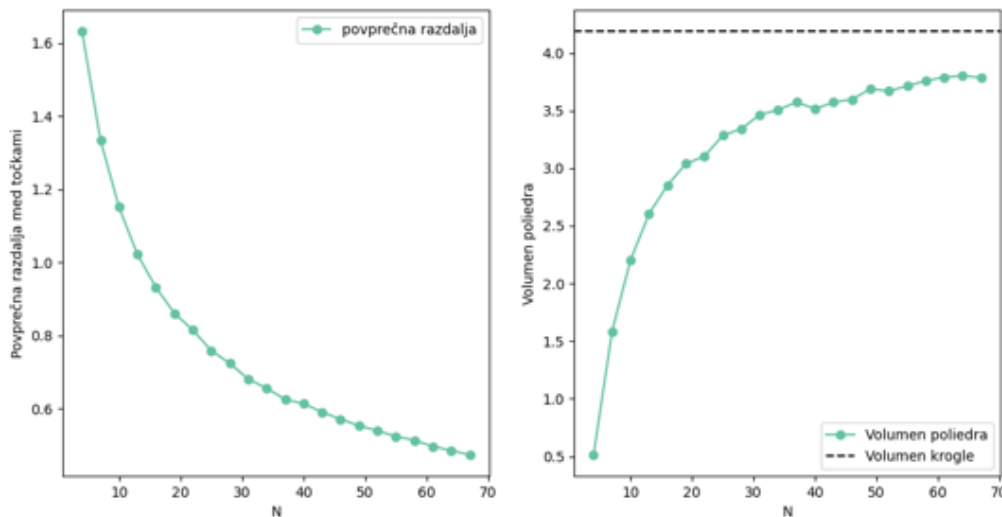
Najprej sem izračunal povprečno razdaljo med naboji. Če je  $r_{ij}$  razdalja med nabojema  $i$  in  $j$ , potem povprečna dolžina stranice med sosednjima nabojema  $d_{avg}$  zapišemo kot:

$$d_{avg} = \frac{1}{N_r} \sum_{\langle i,j \rangle} r_{ij},$$

kjer je  $N_r$  število robov poliedra, ki jih določi `ConvexHull`. Rezultati kažejo, da povprečna dolžina stranic med sosednjimi nabojema konvergira proti nič, ko  $N$  narašča, kar je prikazano na sliki 4. To je pričakovano, saj se ob povečanju števila nabojev njihova razporeditev na sferi bolj enakomerno približuje kontinuumu.

Drugi pomemben parameter, ki sem ga analiziral, je volumen poliedra. Ta volumen se lahko izračuna neposredno z uporabo funkcije `ConvexHull`, ki vrača volumen konveksne lupine. Za poliedre, ki nastanejo pri majhnem  $N$ , je razlika med volumnom poliedra in volumnom krogle večja. Vendar pa, ko  $N$  narašča, volumen poliedra  $V_P$  konvergira proti volumnu krogle, ki je:

$$V_k = \frac{4}{3}\pi r^3,$$



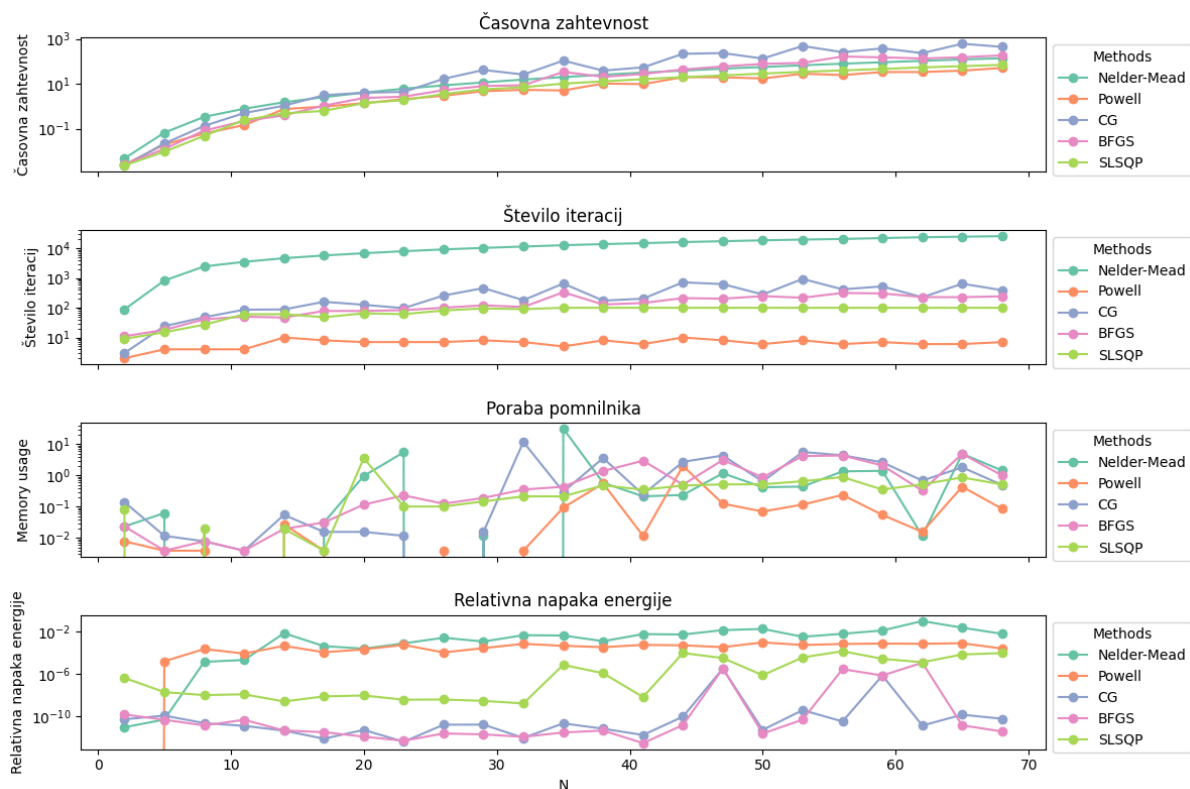
Slika 4: [LEVO] Povprečna razdalja med sosednjimi naboji na sferi, ko gre  $N \rightarrow \infty$  gre krivulja proti 0 [DESNO] Volumen poliedra, ki ga objemajo naboji na sferi. Ko gre  $N \rightarrow \infty$  limitira proti volumnu krogle

Slika 4 prikazuje, da povprečna razdalja med točkami konvergira proti 0 in volumen poliedrov konvergira proti volumnu krogle, kar je nekako intuitivno tudi za pričakovati.

### 3 Analiza različnih algoritmov minimizacije

Kot že omenjeno v poglavju 2, ima *scipy*-jeva metoda *minimize* na voljo več različnih minimizacijskih algoritmov. Splošno znano je, da so nekateri algoritmi boljši za ene probleme nekateri pa za druge. V tem poglavju sem pogledal, kako različni algoritmi minimizirajo Thomsonov problem. Pri analizi sem gledal 5 različnih algoritmov in sicer "Nelder-Mead", "Powell", "CG", "BFGS" in "SLSQP".

Algoritme sem primerjal prek časovne zahtevnosti, števila iteracij, natančnosti in porabe spomina. Natančnost sem izračunal tako, da sem na Wikipediji našel tabelirane vrednosti energije za različne  $N$  [3] in nato izračunal relativno napako  $|E_{wiki} - E|/E_{wiki}$ . Za izračun porabe spomina, pa sem si pomagal s knjižnico *psutil* in funkcijo *memory\_info*.



Slika 5: 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.

Iz slike 5 lahko razberemo, da je najpočasnejša metoda "CG", najhitrejša pa "Powell" in "SLSQP", pri  $N = 70$  je razlika med najhitrejšo in najpočasnejšo metodo že reda skoraj  $10^2$ . "Nelder-Mead" metoda je najpočasnejša verjetno zato, ker za uspešno minimizacijo potrebuje bistveno več iteracij kot ostale metode. Zanimivo pa je, da kljub temu ne porabi kaj več spomina računalnika kot ostale metode.

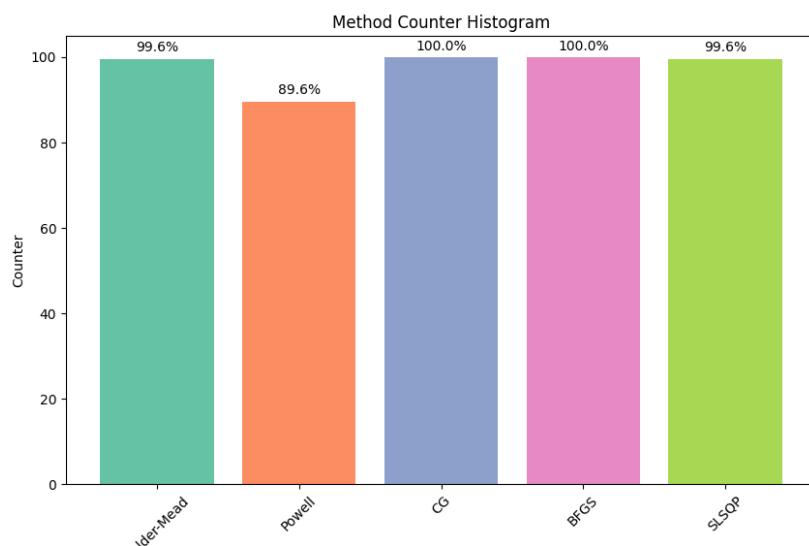
Prav tako iz zadnjega grafa na sliki 5 lahko razberemo, da so vse metode precej natančne pri izračunu energije, kjer je največja relativna napaka  $10^{-2}$ . Vendar pa je med samimi metodami razlika v relativni napaki tudi do  $10^8$ . Prav zaradi tega razloga sem v nadaljevanju uporabljal metodo "BFGS" in "SLSQP", ki sta se izkazali za najbolj natančni za Thomsonov problem.

### 3.1 Uspešnost minimizacije

Poleg časovne zahtevnosti in porabe spomina pri algoritmihi, ki rešujejo optimizacijske probleme, je izjemno pomembna tudi uspešnost pri iskanju globalnega minimuma. Pri mnogih optimizacijskih metodah, še posebej pri nelinejskih in večdimenzionalnih problemih, kot je Thomsonov problem, obstaja tveganje, da algoritem konvergira k lokalnemu minimumu namesto globalnemu. Lokalni minimumi ustrezajo stabilnim, a suboptimalnim konfiguracijam nabojev, kjer je potencialna energija nižja kot v okolici, vendar ne dosega globalnega minimuma, ki predstavlja energijsko najbolj ugodno konfiguracijo sistema.

Da bi preveril uspešnost algoritma, sem se osredotočil na primer s petimi nabitimi delci ( $N = 5$ ). Za oceno robustnosti algoritma pri iskanju globalnega minimuma sem uporabil več popolnoma naključnih začetnih konfiguracij delcev na površini krogle. Te konfiguracije sem nato optimiziral z algoritmom za minimizacijo potencialne energije. Cilj je bil preveriti, ali se končna potencialna energija, ki jo izračuna algoritem, ujema s tabelirano vrednostjo za globalni minimum.

Izračunano energijo  $E_{\text{calc}}$  sem primerjal s tabelirano vrednostjo  $E_{\text{tab}}$ , ki je znana za  $N = 5$ . Globalna energija za  $N = 5$  je  $E_{\text{tab}} = 6.474691495$ .



Slika 6: Uspešnost minimizacije različnih metod za 500 ponovitev pri 5 nabojih na sferi. Začetni pogoji so izbrani naključno.

Izkaže se, da je najmanj zanesljiva metoda "Powell-ova metoda", saj je pravilno napovedala energijo le 90% časa, medtem ko so bile ostale metode skoraj 100% zanesljive. Razlog vidim pri tem, da ima "Powellova" metoda najmanj iteracij pri minimizaciji problema, zato lahko večkrat naleti na lokalni minimum namesto na globalnega.

### 3.2 Dipolni moment

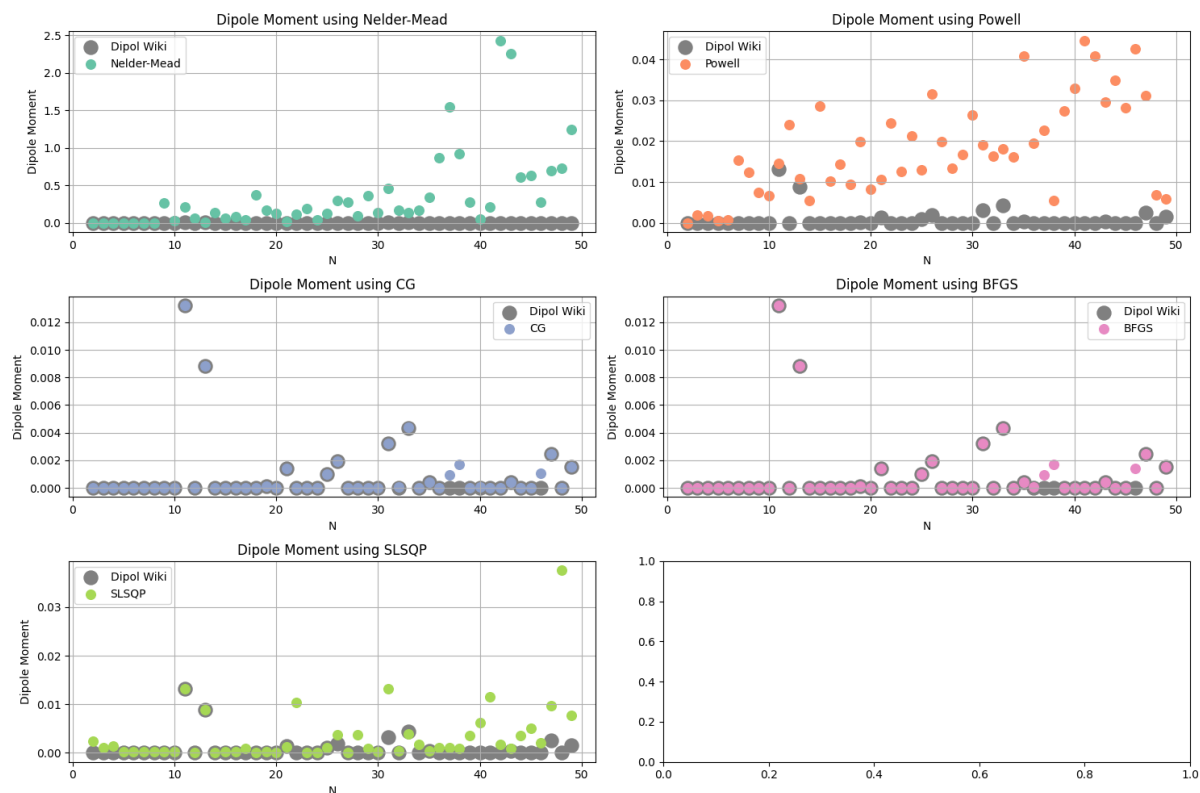
Za različne metode minimizacije sem poleg energije preučil tudi dipolne rešitve Thomsonovega problema. Dipolni momenti predstavljajo pomembno fizikalno lastnost sistema in so prav tako tabelirani na spletu [3]. Dipolni moment  $P$  nam omogoča vpogled v asimetričnost razporeditve nabojev na sferi, saj je v popolnoma simetričnem primeru dipolni moment enak nič.

V brezdimenzijski obliki se dipolni moment izračuna kot preprosta vsota radijevih vektorjev za vsak posamezen naboj v sistemu:



$$P = \sum_i \mathbf{r}_i,$$

kjer je  $\mathbf{r}_i$  radijev vektor  $i$ -tega naboja.

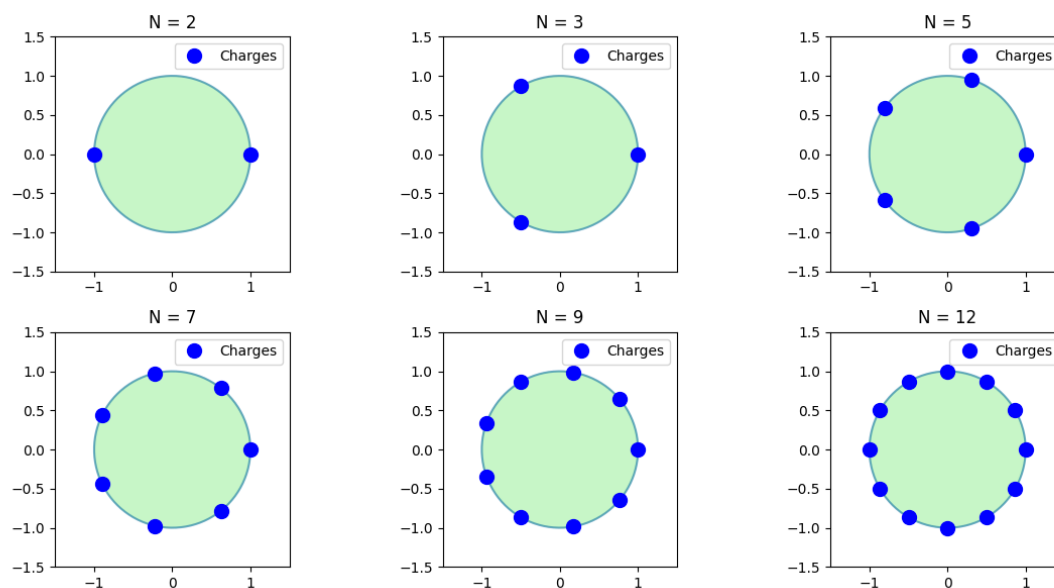


Slika 7: Brezdimenzijski dipolni moment rešitev, pridobljenih preko različnih metod, v odvisnosti od števila nabojev  $N$ . Rešitve so označene z manjšimi pikami, podatki iz Wikipedije pa z večjimi zelenimi.

Čeprav bi pričakovali, da bo v popolnoma optimizirani konfiguraciji  $P = 0$ , lahko zaradi numeričnih napak ali omejitev algoritmov dipolni moment niha okoli ničle.

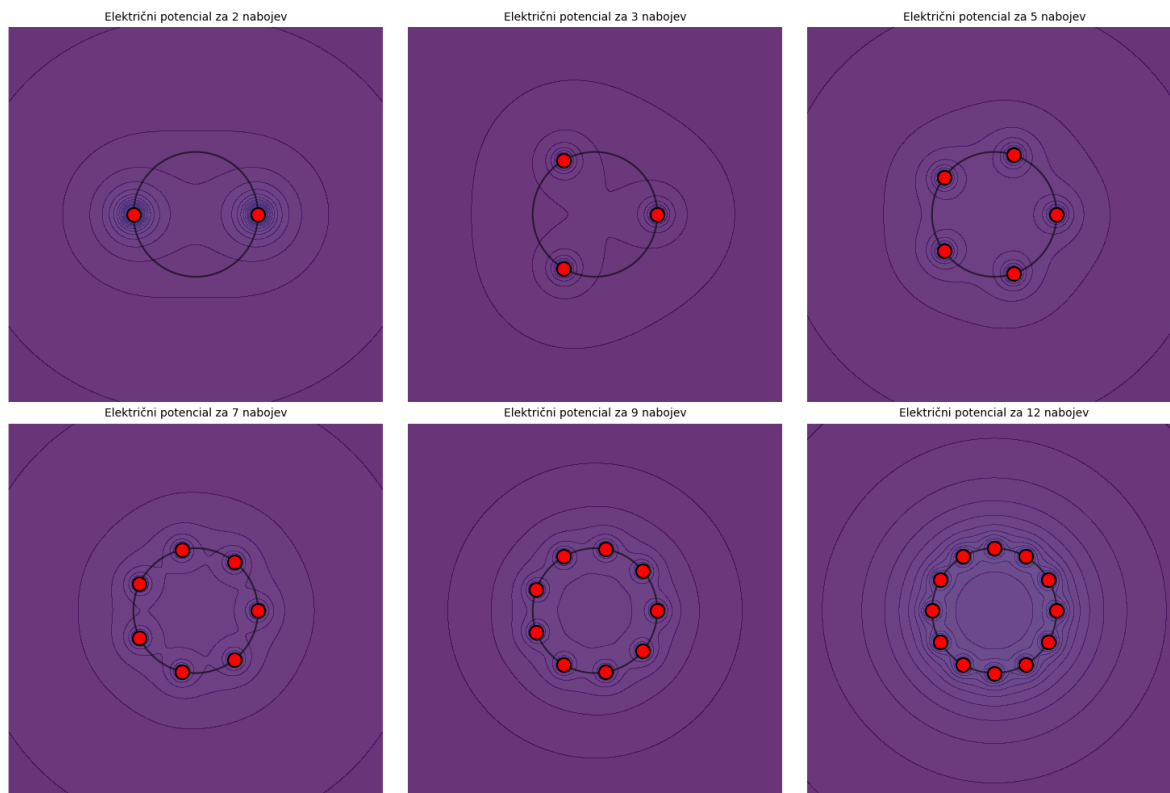
## 4 Dodatno: Geometrija 2D kroga

Za konec analize Thomsonovega problema, pa sem se spustil še eno dimenzijo nižje. Pogledal sem, kako se naboji razporedijo na dvodimenzionalnem krogu. Zgodba je v dveh dimenzijah precej podobna kot v treh, izberemo si začetne pogoje, in enačbo potencialne energije minimiziramo z funkcijo *minimize*. Za minimizacijo sem uporabljal algoritem "BFGS". Nato pa lahko narišemo grafe oz. sheme.



Slika 8: Rešitve Thomsonovega problema za različno število nabojev ( $N = 2, 3, 5, 7, 9, 12$ ) na krogu v dveh dimenzijah. Za minimizacijo je uporabljen algoritem "BFGS"

Kot bi rekli po intuiciji, so se naboji razporedili po robu kroga, kar je tudi globalni minimum. Vendar pa sem opazil, da so se predvsem pri višjih  $N$ , naboji razporejali v lokalne minimume, kjer niso bili vsi na krožnici. Tega nisem šel dlje raziskovat, saj mi je zmanjkalo časa. Mi je pa uspelo za zgoraj narisane konfiguracije izračunati in prikazati električno polje  $N$  nabojev na krogu.



Slika 9: Električno polje rešitev Thomsonovega problema za različno število nabojev ( $N = 2, 3, 5, 7, 9, 12$ ) na krogu v dveh dimenzijah. Za minimizacijo je uporabljen algoritem "BFGS"

## 5 Optimalna vožnja skozi semafor

V drugem delu, pa ponovno rešujemo problem semaforjev, ki je v principu zastavljen enako, kot pri prvi domači nalogi tega predmeta. Razlika je le ta, da moramo tokrat problem diskretizirati in ga rešiti z minimizacijskimi algoritmi. Pri nalogi sem uporabil algoritem *SLSQP*. Lagrangian, ki ostaja enak kot v prvi domači nalogi diskretiziramo s pomočjo trapezne formule in dobimo

$$F = \frac{y_i^2}{2\Delta\tau} \sum_{i=2}^{N-1} \frac{(y_i - y_{i-1})^2}{\Delta\tau} + \frac{1}{2} \frac{(y_N - y_{N-1})^2}{\Delta\tau} \quad (5)$$

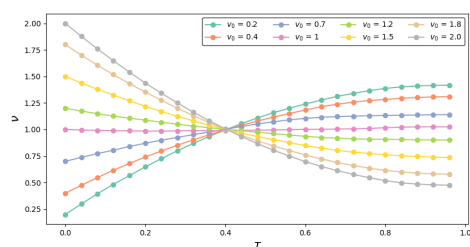
Pogoju o prevoženi poti pa zadostimo z vezjo

$$F_1 = \frac{1}{2} y_1 \Delta\tau + \sum_{i=2}^{N-1} y_i \Delta\tau + \frac{1}{2} y_N \Delta\tau \quad (6)$$

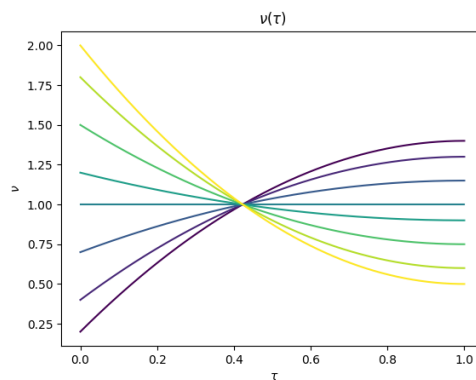
Funkcija, ki jo minimiziramo je potem

$$F_{\min} = F + F_1 \quad (7)$$

Ker je to še vseeno neka primerjava s prvo domačo nalogo sem pogledal 3 enake primere, kot sem jih pogledal pri prvi nalogi. Najprej sem pogledal primer prostega robnega pogoja tj. pri semaforju je odvod hitrosti enak 0.

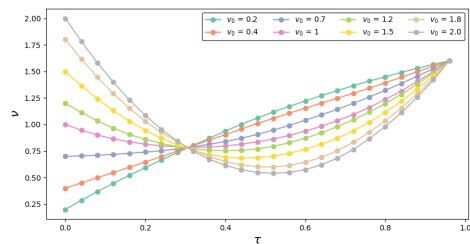


Slika 10: Graf brezdimenzijske hitrosti v odvisnosti od brezdimenzijskega časa pri rešitvi problema z prostimi robnimi pogoji. Problem je bil diskretiziran in minimiziran z metodo "BFGS"

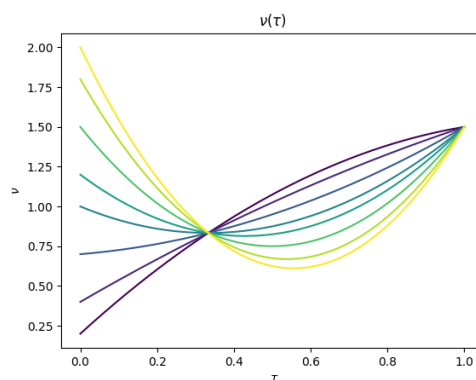


Slika 11: Graf brezdimenzijske hitrosti v odvisnosti od brezdimenzijskega časa pri analitični zvezni rešitvi problema z prostimi robnimi pogoji.

Naslednji primer je bil primer, kjer sem pogledal, kaj se zgodi če je pri semaforju radar ali z drugimi besedami, da je končna hitrost omejena

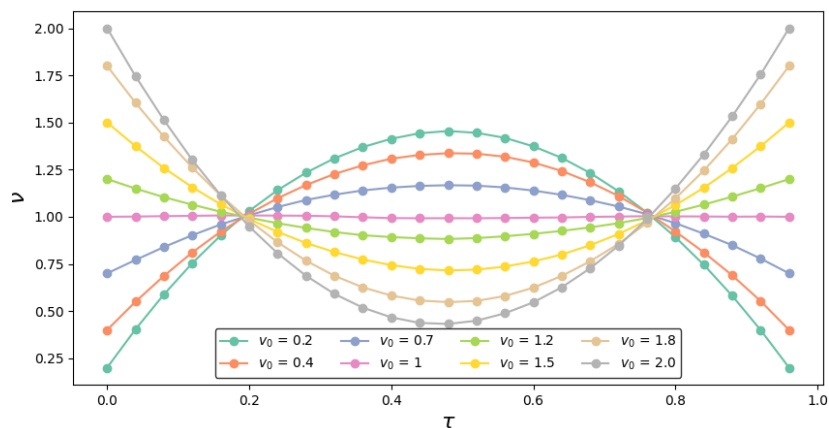


Slika 12: Graf brezdimenzijske hitrosti v odvisnosti od brezdimenzijskega časa pri rešitvi problema z določeno končno hitrostjo. Problem je bil diskretiziran in minimiziran z metodo "BFGS"



Slika 13: Graf brezdimenzijske hitrosti v odvisnosti od brezdimenzijskega časa pri analitični zvezni rešitvi problema z določeno končno hitrostjo

Vidimo, da se oba primera dobro, če ne skoraj popolnoma sovpadata z narisano analitično zvezno rešitvijo. Za na konec pa sem še narisal primer, kjer sta začetna in končna hitrost enaki. Pričakovano, dobimo bolj kompletno parabolo, kot pri prvih dveh primerih.

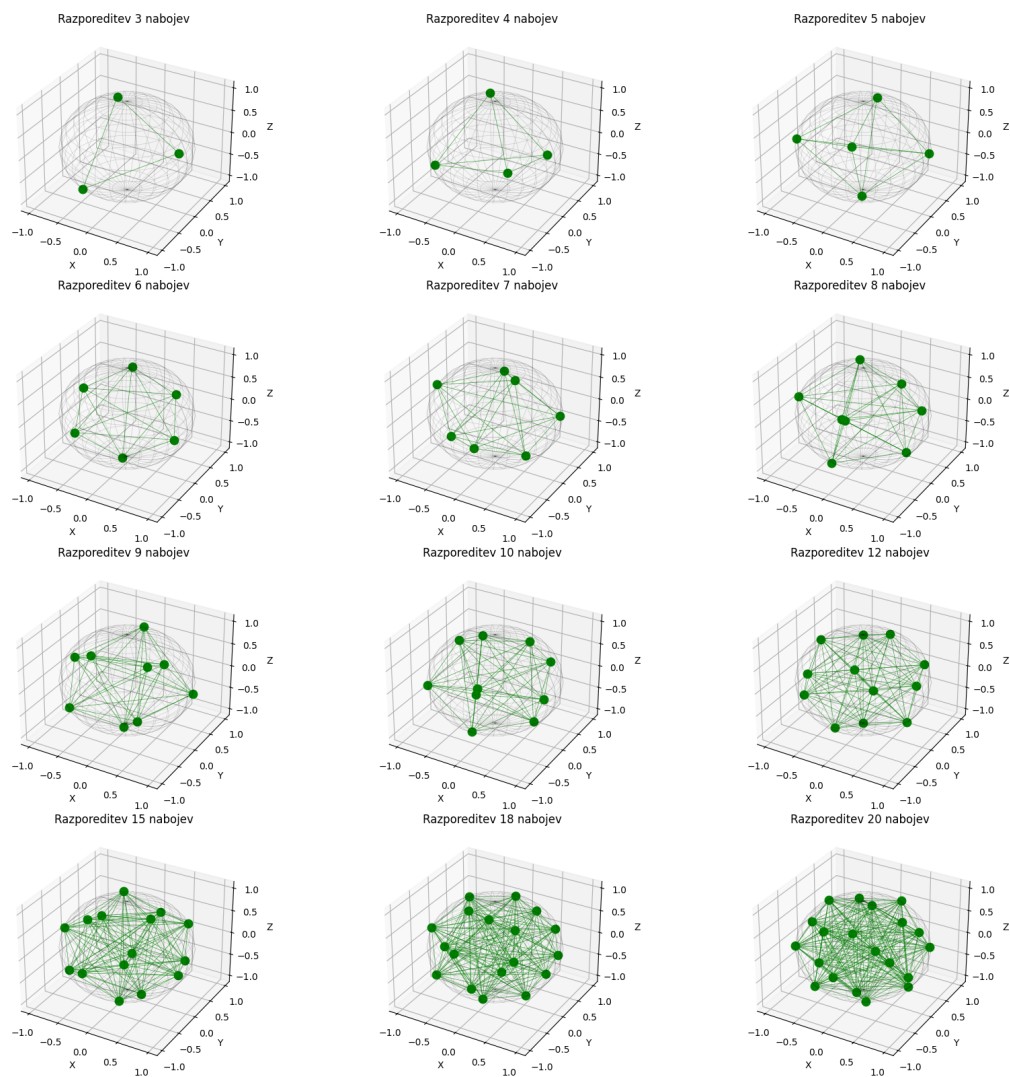


Slika 14: Graf brezdimenzijske hitrosti v odvisnosti od brezdimenzijskega časa pri rešitvi problema z določeno končno hitrostjo, ki je enaka začetni. Problem je bil diskretiziran in minimiziran z metodo "BFGS"

## Literatura

- [1] SciPy Developers. *scipy.optimize.minimize*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>. Accessed: 2024-10-23. 2024.
- [2] SciPy Developers. *scipy.spatial.ConvexHull*. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.ConvexHull.html>. Accessed: 2024-10-23. 2024.
- [3] Wikipedia contributors. *Thomson problem*. [https://en.wikipedia.org/wiki/Thomson\\_problem](https://en.wikipedia.org/wiki/Thomson_problem). Accessed: 2024-10-23. 2024.

## A Thomsonov problem za različne $N$



Slika 15: Rešitve Thomsonovega problema za različno število nabojev ( $N = 2, 4, 5, 6, 7, 8, 9, 10, 11, 15, 18, 20$ ) na krogli. Z zelenimi pikami narisani naboji, zelene tanke črte pa so stranice poliedra, ki ga točke objemajo. Sfera je narisana z tanko sivo obrobo. Za minimizacijo je uporabljen algoritem "Nelder-Mead"