

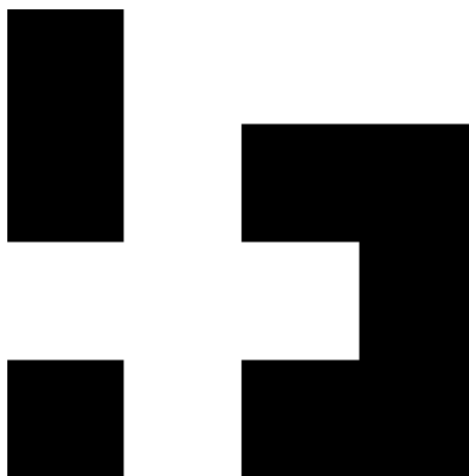
206: Parcialne diferencialne enačbe: lastne rešitve

Peter Rupnik

31. marec 2021

1. naloga

Določi nekaj najnižjih lastnih frekvenc in lastnih nihanj kvadratne opne, ki je odebeljena na večjo ploskovno gostoto povesod, kjer je slika spodaj senčena sivo. Prevedi problem na sistem diferenčnih enačb. Kaj pa, če je senčeno območje lažje? Oglej si odvisnost spektra od gostote sivega polja.



Poleg direktne diagonalizacije preskusi še metodo z inverzno iteracijo. Ker je matrika redka in nas zanima le prvih nekaj lastnih vrednosti in vektorjev, lahko uporabiš tudi knjižnice za delo z redkimi matrikami. Konvergenca, hitrost in natančnost metod preskusi na neobteženi opni.

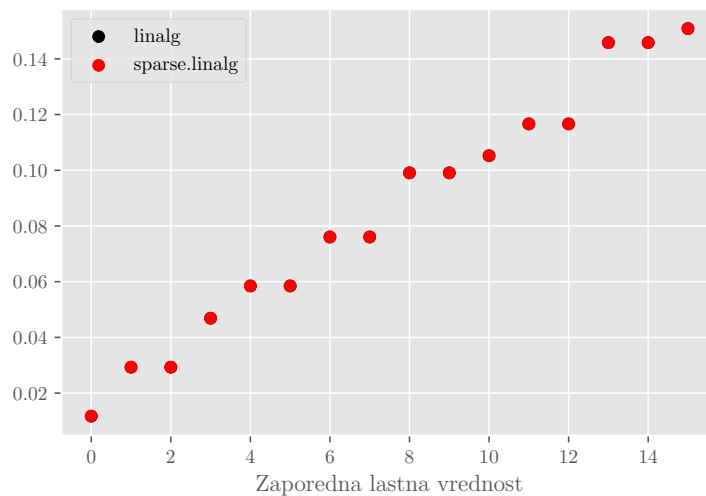
1.1 Tipanje problema

Pričel sem z neokrnjeno, kvadratno *tabulo raso*. Uveril sem se, da znam poiskati lastne nihajne načine in pripadajoče lastne vrednosti. Po pripravi matrike A sem ugotovil, da metoda `scipy.linalg.eigh` deluje odlično, metode iz naprednejšega modula

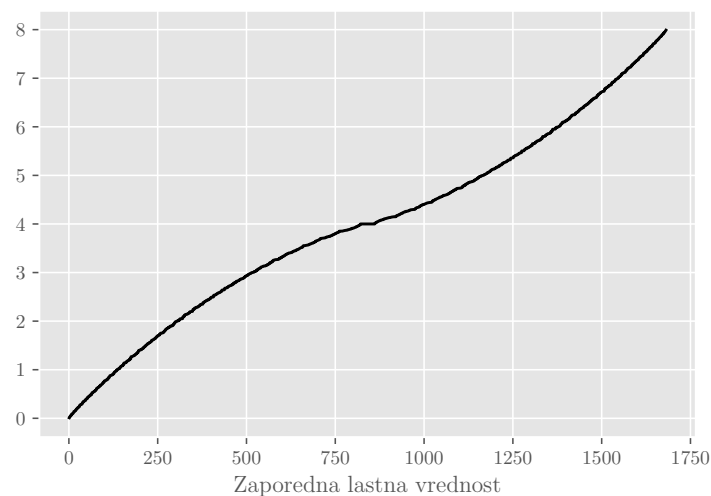
`scipy.sparse.linalg` pa so malenkost zahtevnejše, vendar s pravo optimizacijo parametrov povrnejo ekvivalentne rezultate.

Prvih nekaj lastnih vrednosti prikazujem spodaj. Opazimo, da lastne vrednosti ležijo druga na drugi, ker pa imajo vsake oči svojega malarja, sem preveril še numerično ekvivalenco obeh vektorjev lastnih vrednosti:

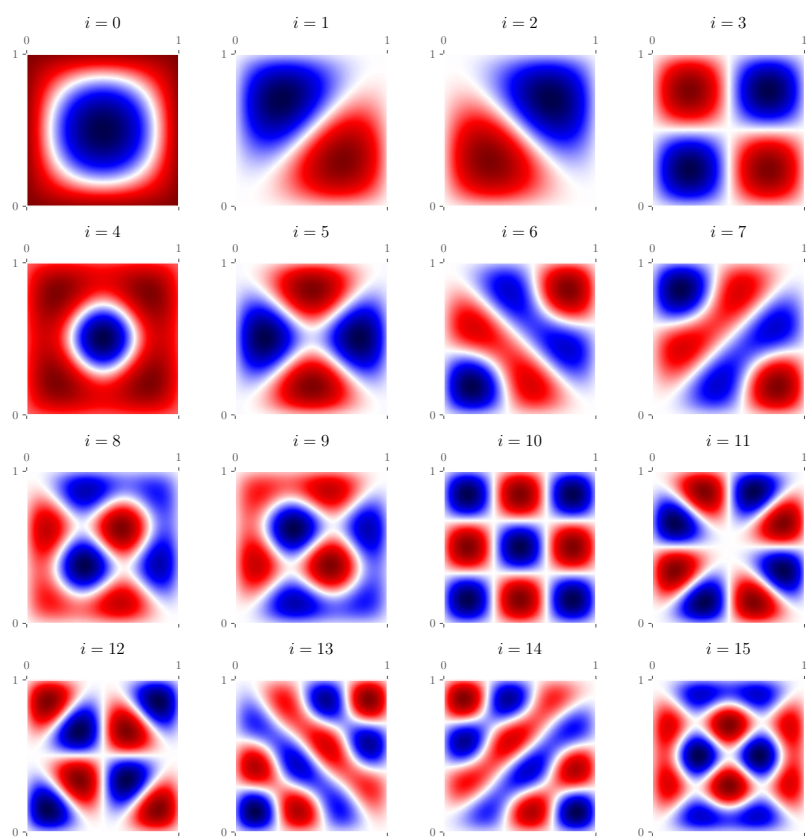
```
np.allclose(eigenvalues_sparse[:16], eigenvalues[:16])  
True
```



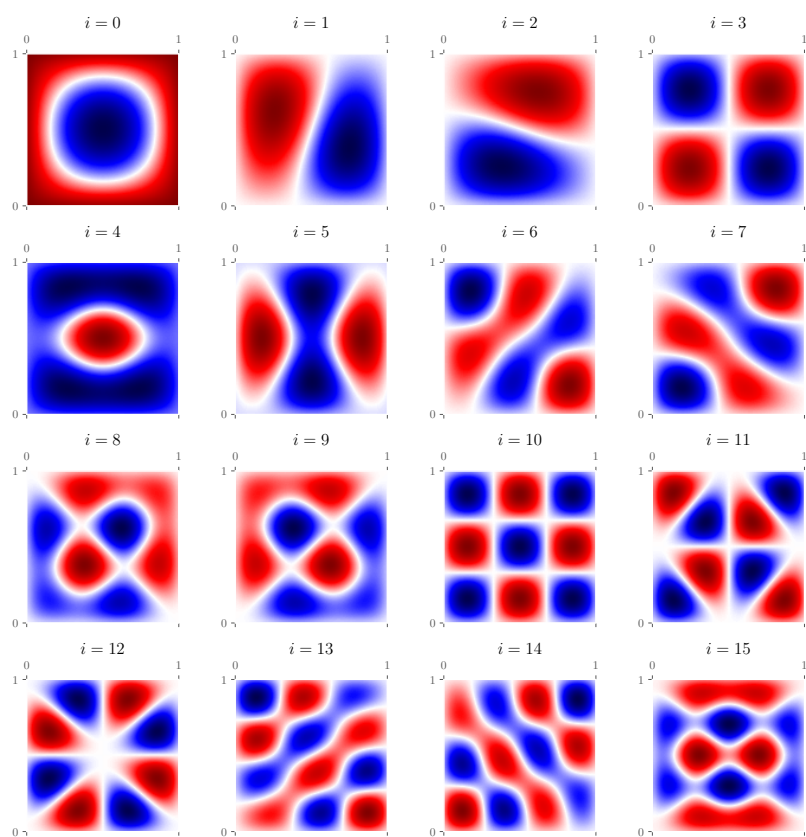
Če nadaljujemo raziskovanje lastnih vrednosti vse do višjih, dobimo ponovno ekvivalenco naračunanih lastnih vrednosti. Izgled polnega spektra prikazujem spodaj:



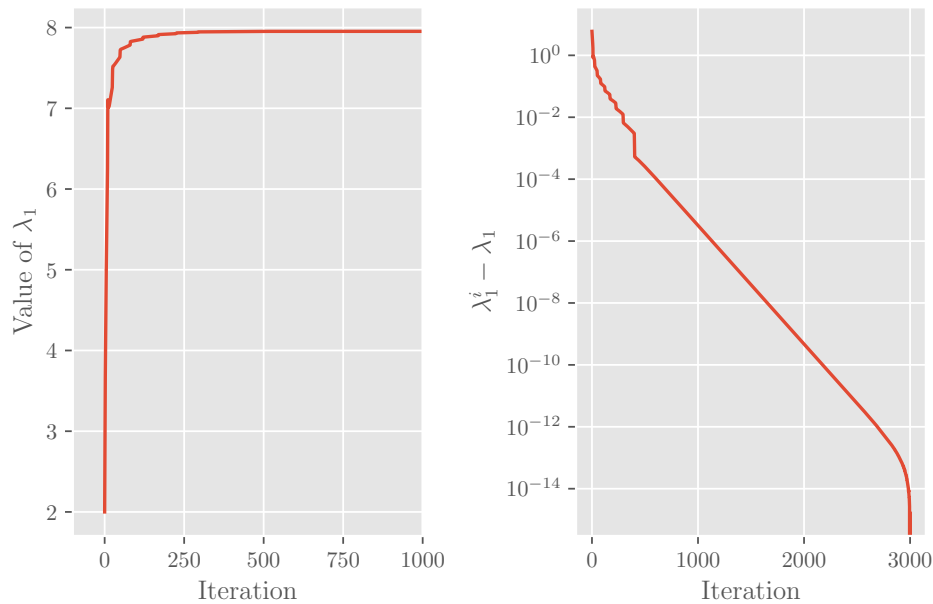
Akadska radovednost veli, da izrišemo vsaj nekaj osnovnih nihajnih načinov, ki jih porodita obe metodi. Za 'klasično' metodo, ki ne upošteva posebnih odlik vhodne matrike, dobimo sledečo sliko.



Če uporabimo metodo iz modula `scipy.sparse`, dobimo precej podobno sliko:

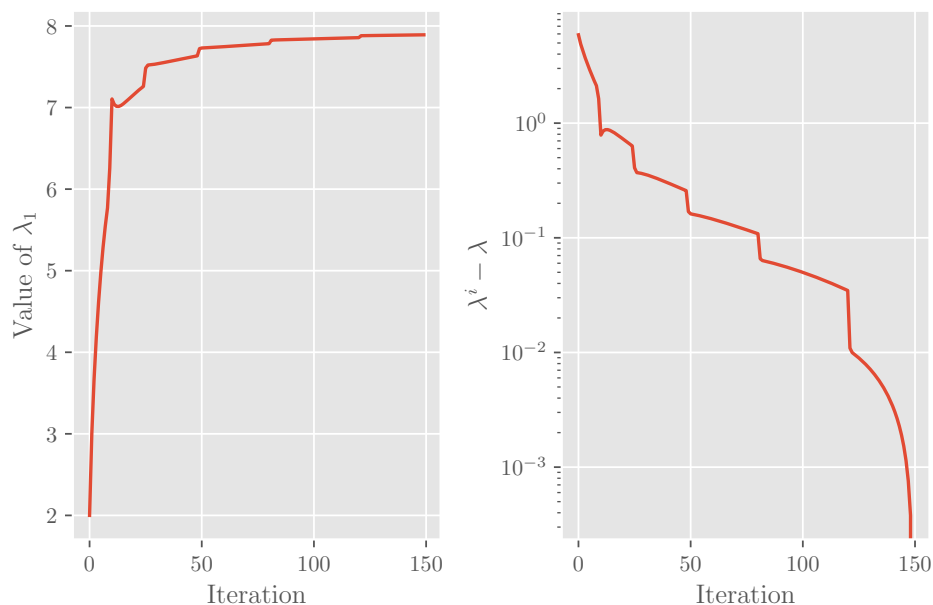


Nadaljeval sem z implementacijo potenčne metode. Sprva mi je nekaj težav predstavljala normalizacija lastnih vektorjev, a mi je kasneje uspelo:

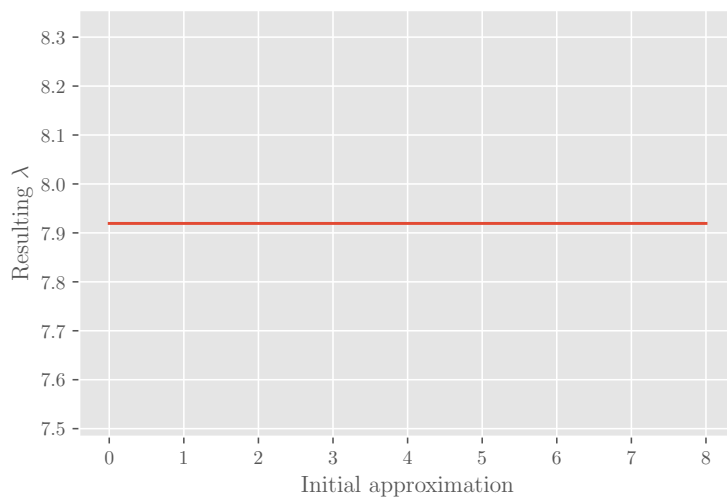


Naslednji korak je nadgradnja metode za konvergenco k poljubni lastni vrednosti. Opazim, da zaradi iskanja inverza matrike $A - \sigma I$ potrebujem mnogo več časa za enako število iteracij. Optimizacija z knjižnjico `numba.jit` ni uspela. Pri generaciji spodnje slike sem uporabil začetni približek $\lambda = 8$ in dosegel isto lastno vrednost kot prej. Poleg potrebe po invertiranju matrike vsako iteracijo algoritem dodatno podaljša tudi dejstvo, da je \mathcal{O} metode po [1] enak

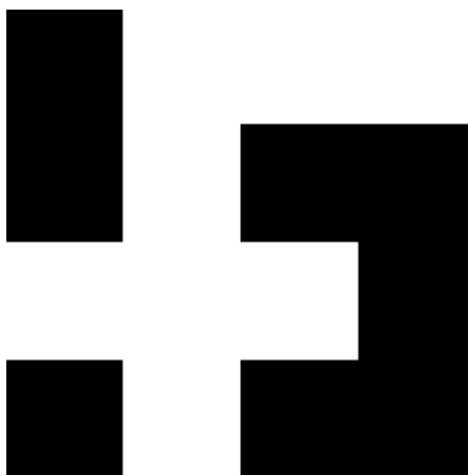
$$\left| \frac{\sigma - \lambda_{\text{iskana}}}{\lambda_{\text{naslednja najbližja}} - \lambda_{\text{iskana}}} \right|$$



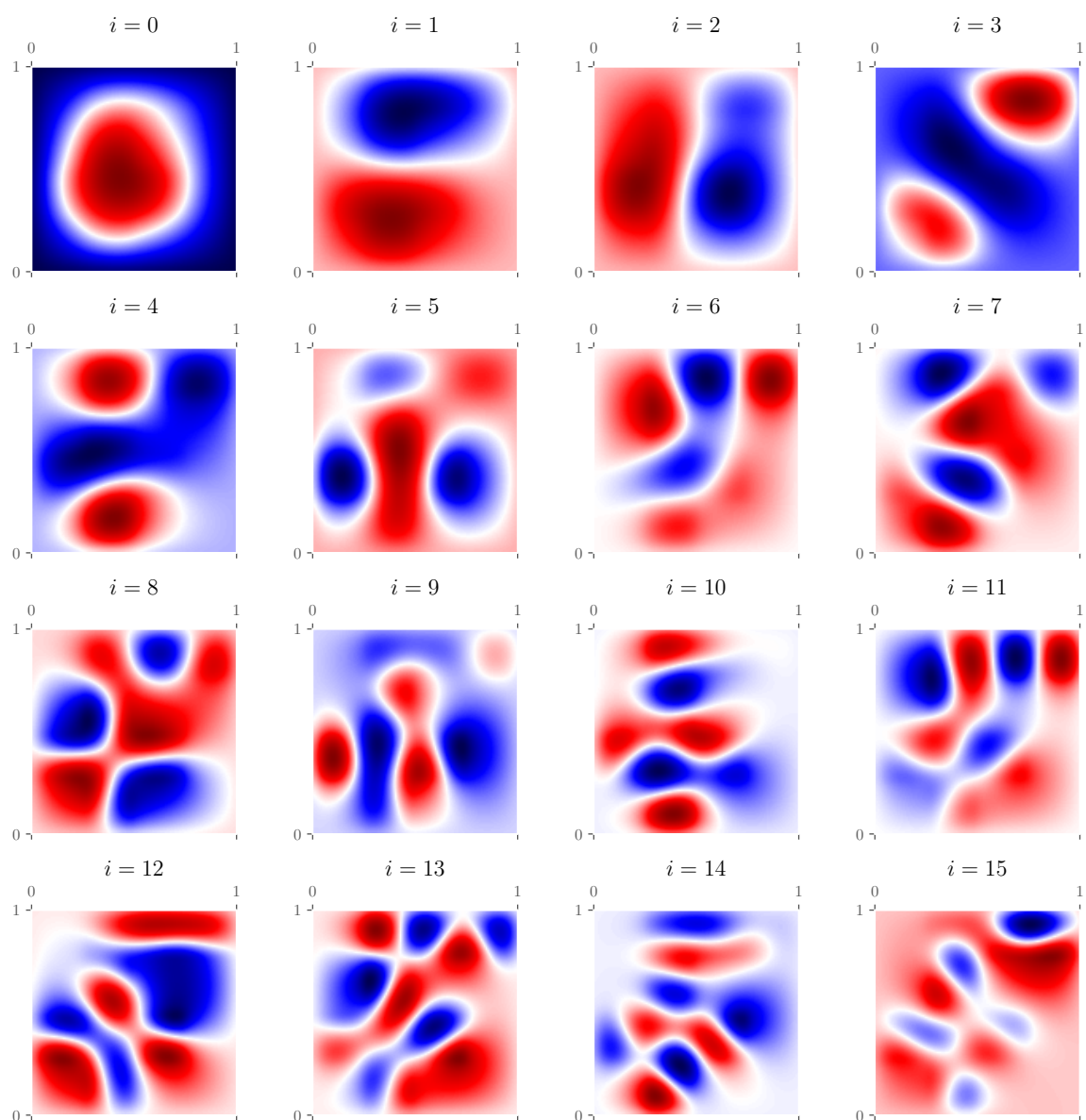
Če inverzno potenčno metodo uporabim na razponu od 0 do maksimalne lastne vrednosti, dobim spodnjo sliko, ki nakazuje, da metoda ne deluje, kot bi si želeli.



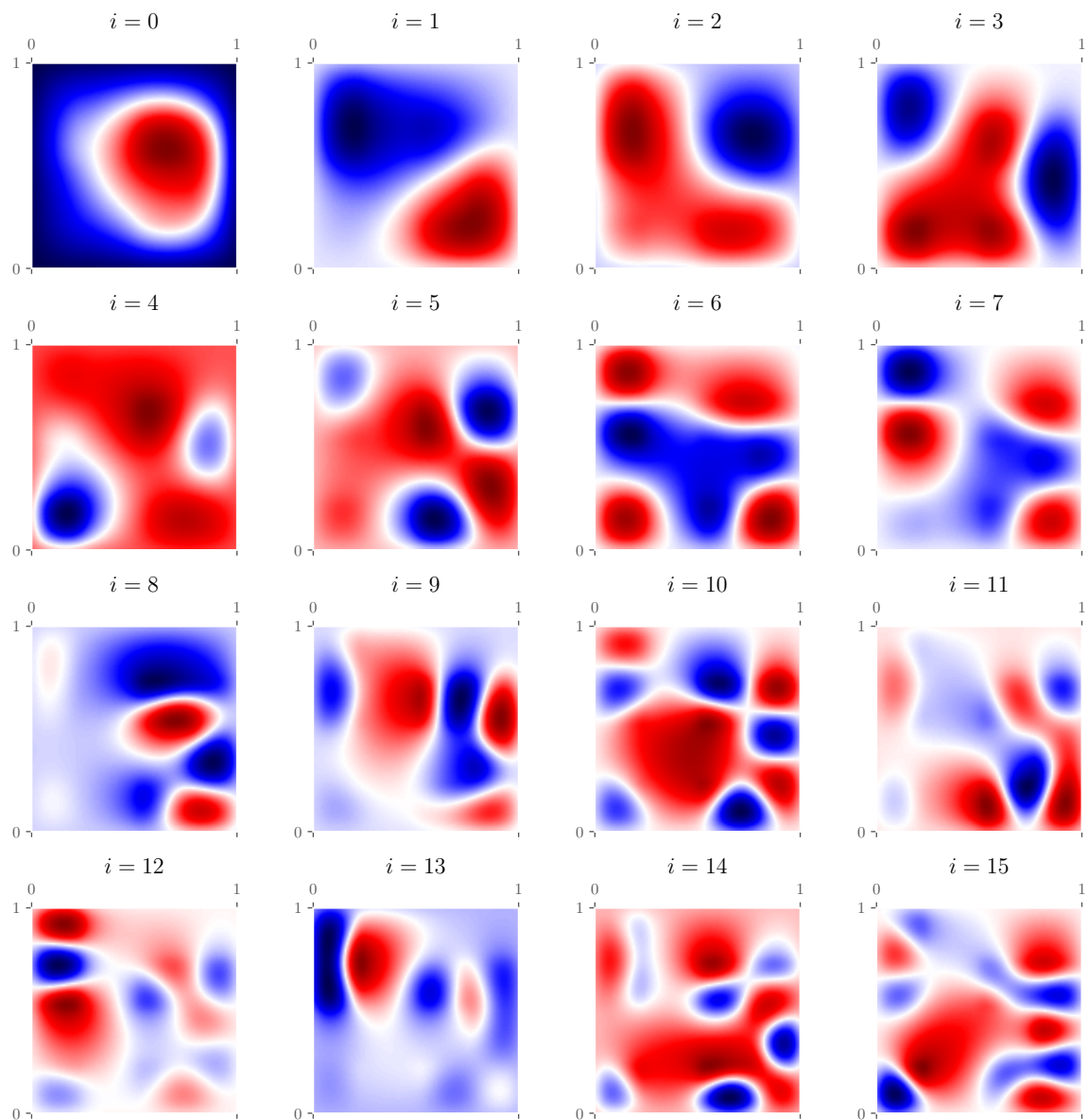
V nadaljevanju sem zgeneriral zahtevano figuro:



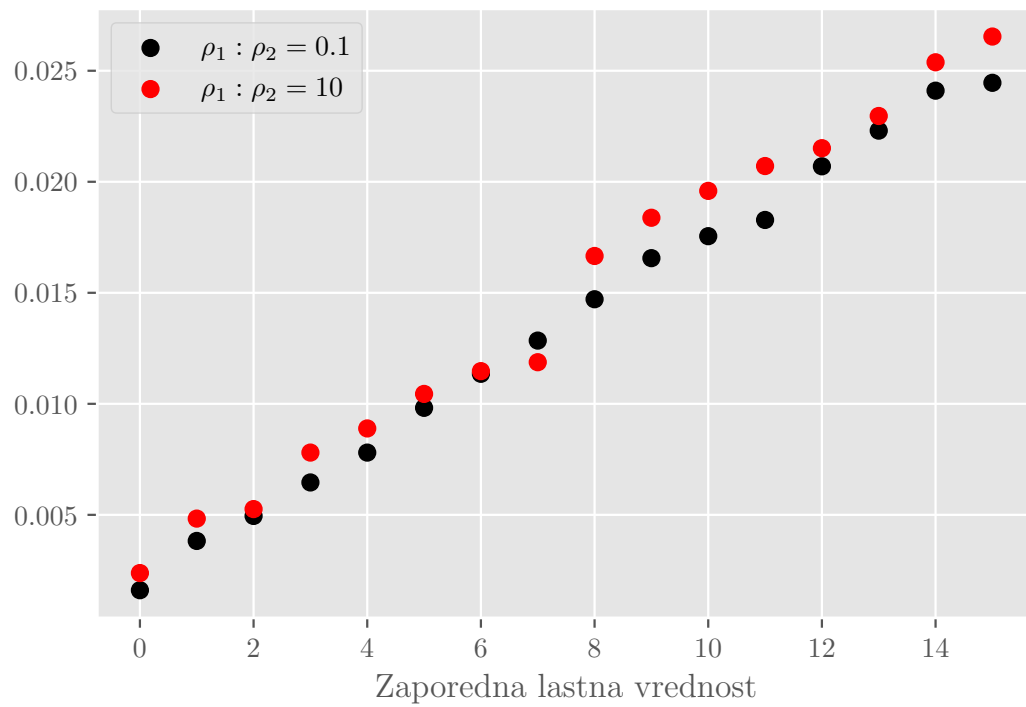
Če dobljeno gostotno matriko $\rho(x, y)$ preoblikujemo v diagonalno matriko, jo lahko uporabimo v metodi `scipy.linalg.eigh` in izračunamo lastne vrednosti in lastne vektorje kot poprej:



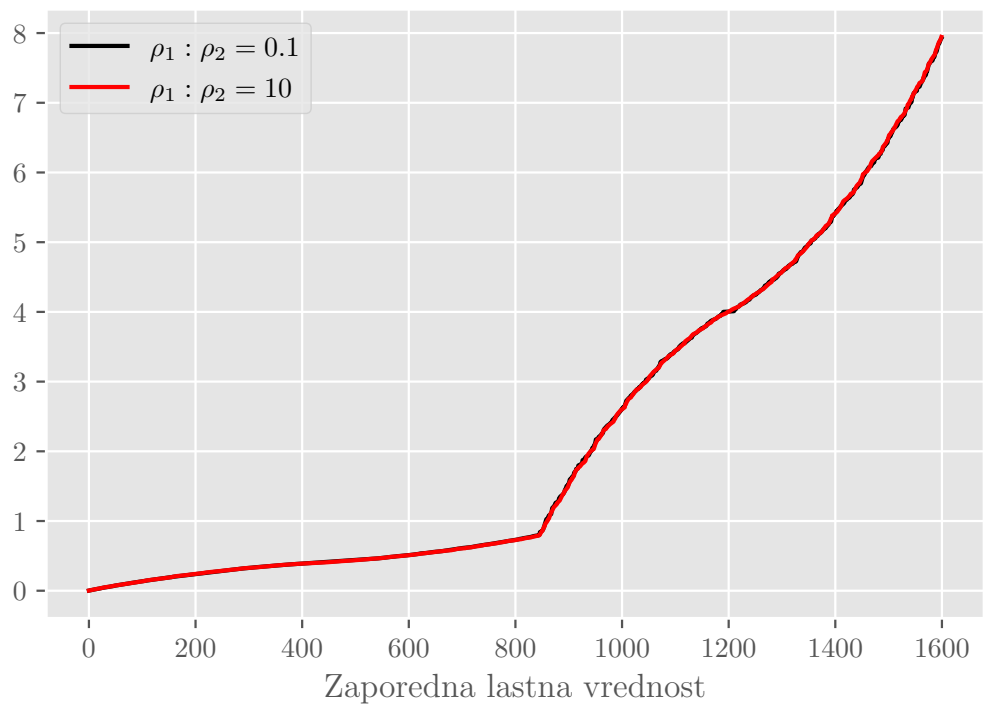
Prikazujem prvih 16 nihajnih načinov, uporabljeni gostoti sta 1 in 10, osenčena območja so tako manj gosta. Če gostoti zamenjam, dobim naslednjo sliko.



Poglejmo si še spekter obeh obtežitev, začeniši z najnižjimi 16 lastnimi vrednostmi.

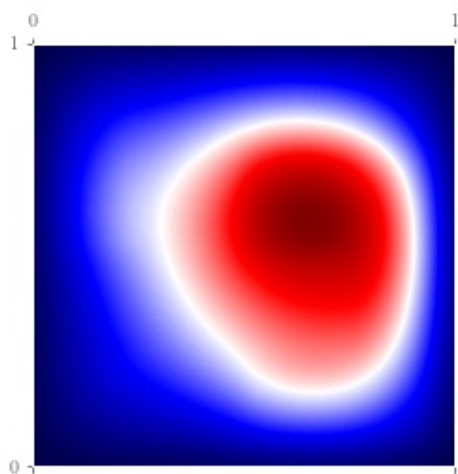


Če pogledamo celoten spekter obeh obtežitev, dobimo spodnjo sliko.

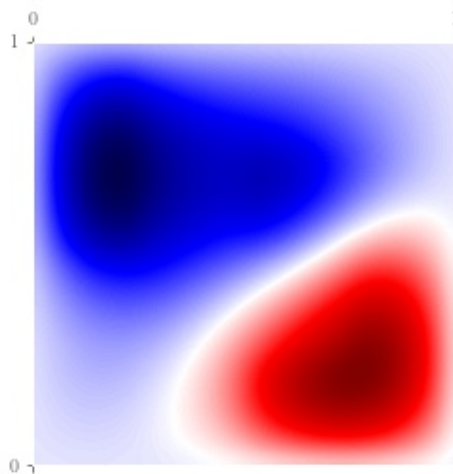


Očitno je, da pride nekje na polovici do diskontinuitete, zato sem si ta prehod pogledal

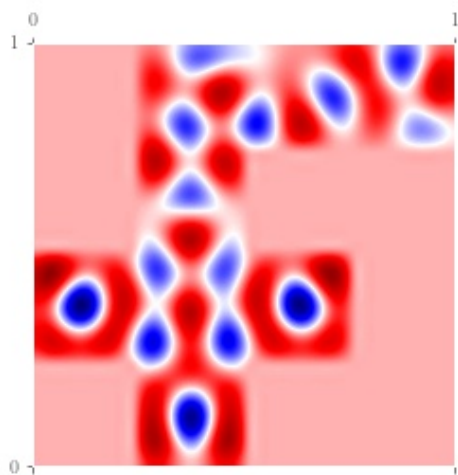
pobližje na še bolj ekstremnem razmerju gostot 1000:1. Najnižja lastna nihanja izgledajo presenetljivo podobna vsem ostalim:



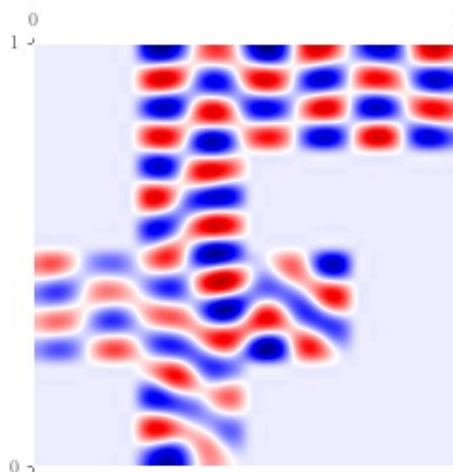
Prvi nihajni način za razmerje gostot 1000:1.



Drugi nihajni način za isto opno.



850. nihajni način za razmerje gostot 1000:1.



901. nihajni način za isto opno.

Postuliram, da zaradi diskontinuitete v spektru pride zaradi kvalitativno različnih nihanj; če prej celotna opna niha kot celota, po neki kritični lastni vrednosti težji deli opne obmirujejo, nihanje pa se z venomer večjimi valovnimi števili razvije v lažjem delu.

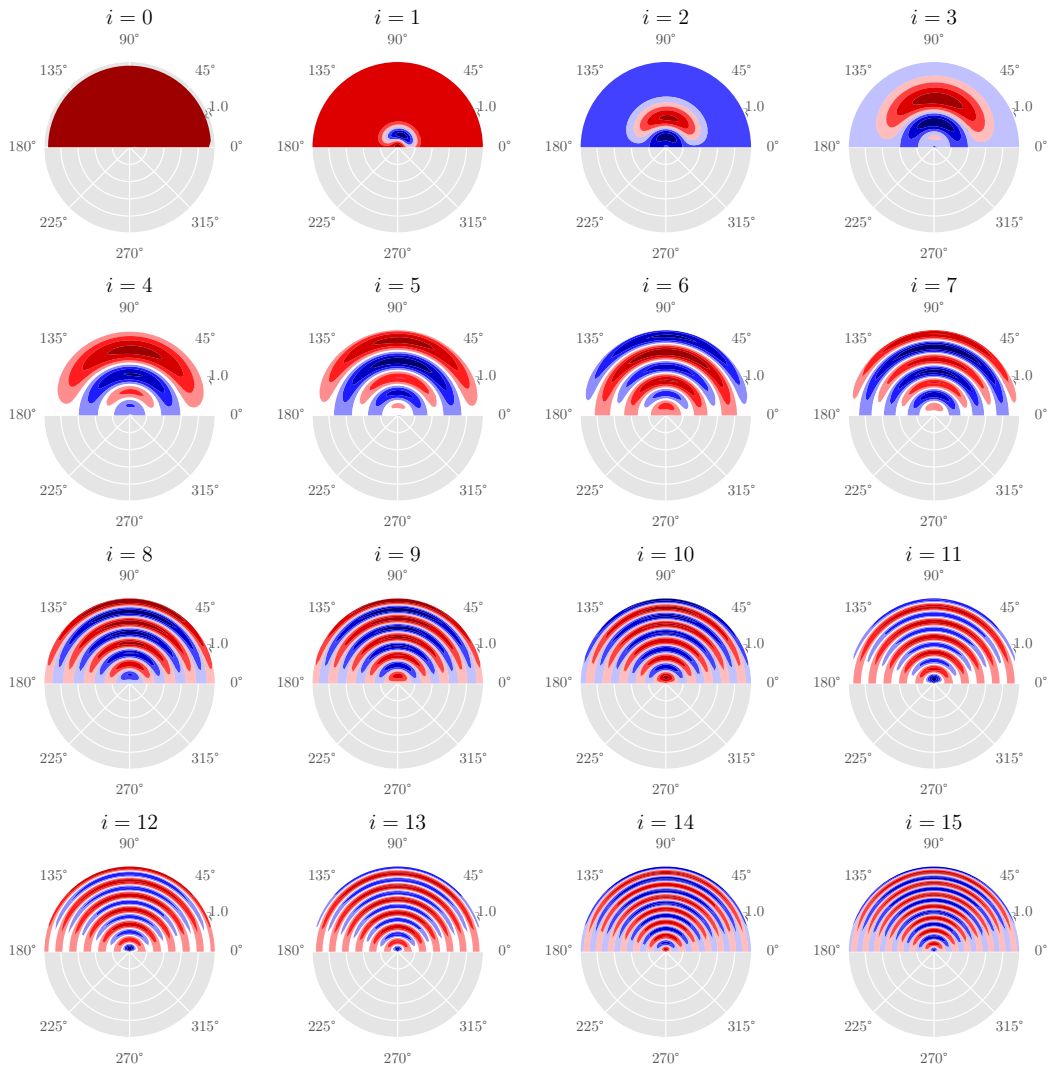
2. naloga

Oceni lastne vrednosti operatorja ∇^2 na problemu lastnih rešitev za polkrog z robnim pogojem I. vrste.

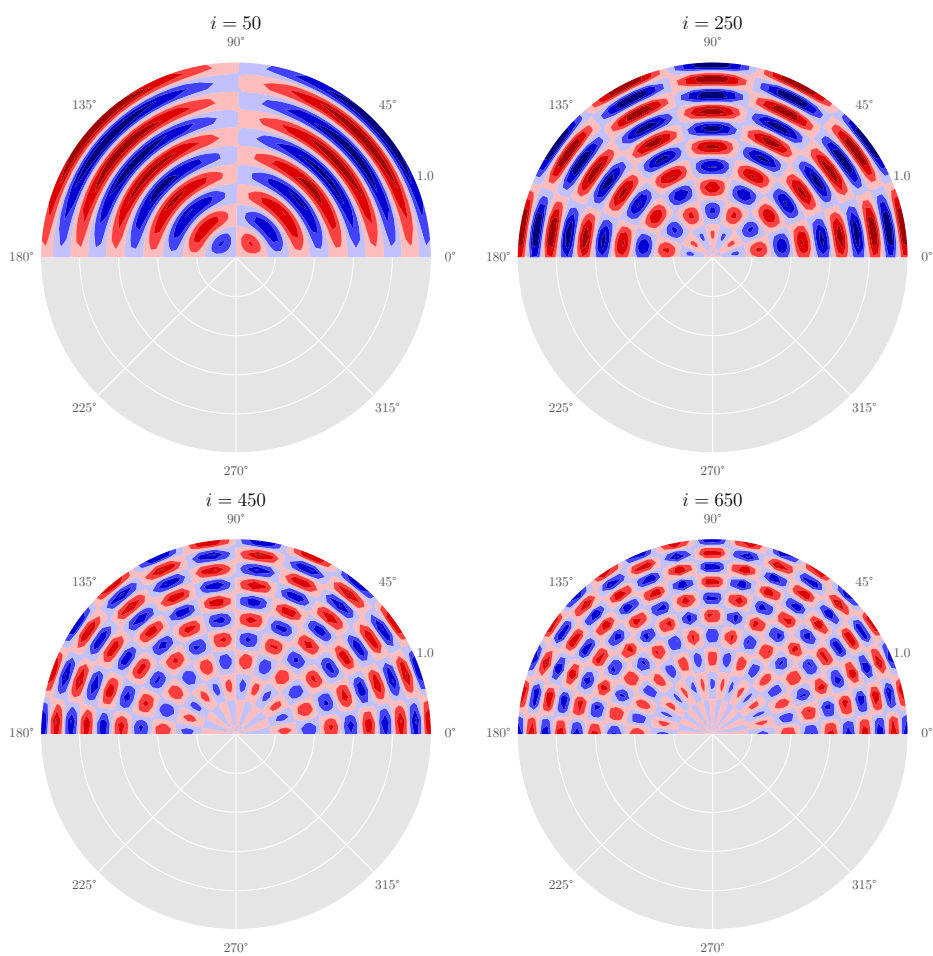
2.1 Reševanje

Ponovno pripravim matriko, ki v diskretni obliki opisuje operator ∇^2 . Tokrat ni asimetrična, zato sem pri uporabi `scipy.sparse.linalg` našel kompleksne rešitve (sicer z imaginarno komponento, primerljivo z $\varepsilon_{\text{machine}}$). Z knjižnjico `scipy.linalg` teh problemov nisem imel, zato sem se posluževal slednje. Na podlagi rezultatov prve naloge pričakujem, da bi se metodi dobro ujemali in bo nadaljnje obnašanje rešitev podobno.

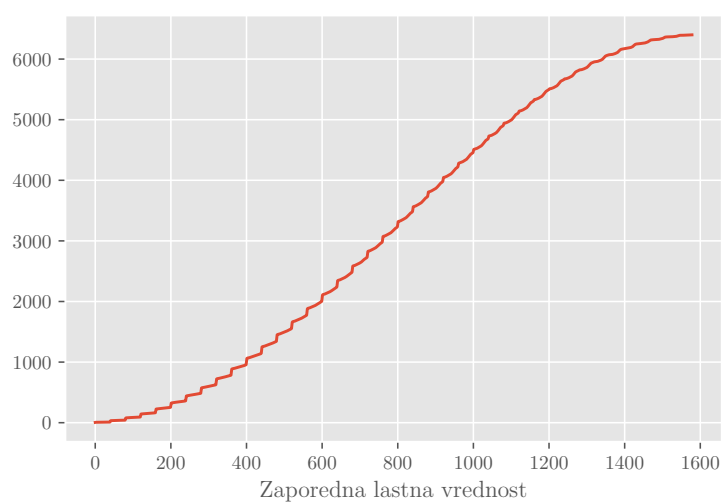
Ko sem lastne vektorje in vrednosti izračunal, sem si jih izrisal v polarni geometriji.



Ker me je skrbelo, zakaj ne vidim večje polarne odvisnosti, sem pogledal tudi višje lastne nihajne načine in se uveril, da dobim pričakovano obnašanje tudi po polarnem kotu.



Spekter matrike, ki opisuje operator ∇^2 , prikazujem spodaj.



Literatura

[1] https://en.wikipedia.org/wiki/Inverse_iteration