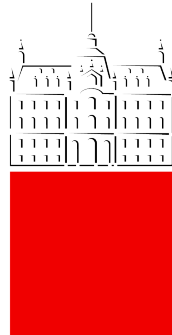


Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



Modelska analiza 2

8. naloga - Metoda končnih elementov: Lastne
rešitve

Študent: Pšeničnik Tomaž

5. maj 2019

Lastne rešitve FEM - Teorija

Z metodo končnih elementov (FEM) želimo poračunati lastne nihajne načine polkrožne opne. Problem je zelo podoben reševanju *Poissonove* enačbe v prejšnji nalogi, le da tokrat rešujemo s pomočjo lastnih vrednosti. Enačba se tokrat glasi

$$\nabla^2 u = -k^2 u, \quad (1)$$

kjer je k lastna vrednost sistema.

Podobno kot dobimo *Poissonovo* enačbo z zahtevo po ekstremu funkcionala $\mathcal{S}(u) = \frac{1}{2} \langle \nabla u, \nabla u \rangle - \langle f, u \rangle$, dobimo tudi enačbo (1) z zahtevo po ekstremu funkcionala

$$\mathcal{S}(u) = \frac{1}{2} \langle \nabla u, \nabla u \rangle - \frac{1}{2} k^2 \langle u, u \rangle, \quad (2)$$

kjer je skalarni produkt definiran na L^2 prostoru kot

$$\langle a, b \rangle = \int a \cdot b \, dxdy.$$

Če zapišemo funkcional (2) kot $\mathcal{S}(u) = \int \mathcal{L}(\nabla u, u) dxdy$, dobimo z *Euler-Lagrangevo* enačbo:

$$\sum_i \frac{\partial}{\partial x_i} \frac{\partial \mathcal{L}}{\partial u_{x_i}} - \frac{\partial \mathcal{L}}{\partial u} = 0$$

ravno enačbo (1). Toda naš cilj ni reševanje E-L enačb, temveč varrirati funkcional (2) po baznih funkcijah, kar je enako iskanju ekstrema funkcionala.

Kot v prejšnji nalogi, tudi tokrat razvijemo iskano funkcijo u po baznih funkcijah

$$u = \sum_{i=1}^N c_i \phi_i$$

in vstavimo v funkcional (2). Dobimo

$$\mathcal{S} = \frac{1}{2} \sum_{ij} \langle c_i \phi_i, c_j \phi_j \rangle - \frac{1}{2} k^2 \sum_{ij} \langle c_i \phi_i, c_j \phi_j \rangle. \quad (3)$$

Kot omenjeno, iščemo ekstrem funkcionala (3) po baznih funkcijah, kar prepoznamo kot sistem linearnih enačb, ki ga zapišemo:

$$\frac{\partial \mathcal{S}}{\partial c_i} = 0 \rightarrow \sum_{ij} (A_{ij} - k^2 B_{ij}) c_j = 0 \quad \forall i. \quad (4)$$

Za bazne funkcije bomo vzeli iste funkcije kot v prejšnji nalogi. Izračun lokalne matrike A_{ij}^k tako poznamo že od prejšnje naloge in je enak:

$$A_{ij}^k = \langle \nabla \phi_i, \nabla \phi_j \rangle = \frac{1}{2|J_k|} (y_{i+1} - y_{i+2}, \quad x_{i+2} - x_{i+1}) \begin{pmatrix} y_{j+1} - y_{j+2} \\ x_{j+1} - x_{j+2} \end{pmatrix}.$$

Za lokalno B_{ij}^k matriko pa velja

$$\langle \phi_i, \phi_j \rangle = \int_{\Delta_k} \phi_i \phi_j dx dy = \int_{\Delta_*} \chi_i \chi_j |J_k| d\xi d\eta,$$

kjer je ϕ_i bazna funkcija trikotnika Δ_k podanega območja in χ_i bazna funkcija referenčnega trikotnika Δ_* , $|J_k| = 2\Delta_k$ pa Jacobijeva determinanta, ki je konstantna in enaka dvakratni ploščini Δ_k . Tako dobimo za integral po baznih funkcijah referenčnega trikotnika

$$B_{ii}^k = \int_{\Delta_*} \chi_i \chi_i |J_k| d\xi d\eta = \frac{|J_k|}{12} \quad \text{in} \quad B_{i \neq j}^k = \int_{\Delta_*} \chi_i \chi_j |J_k| d\xi d\eta = \frac{|J_k|}{24}$$

oziroma, za območje Δ_k

$$B_{ii}^k = \frac{1}{6} \Delta_k \quad \text{in} \quad B_{i \neq j}^k = \frac{1}{12} \Delta_k. \quad (5)$$

Potrebno je sestaviti globalni matriki A in B ter tako poračunati sistem

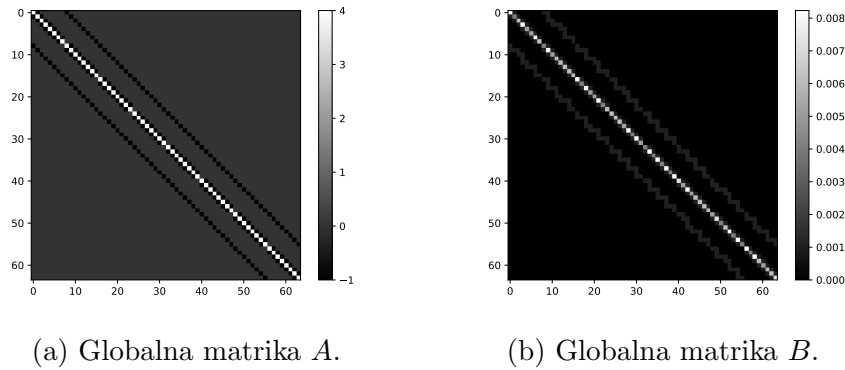
$$A\vec{c} = k^2 B\vec{c},$$

kjer je rešitev

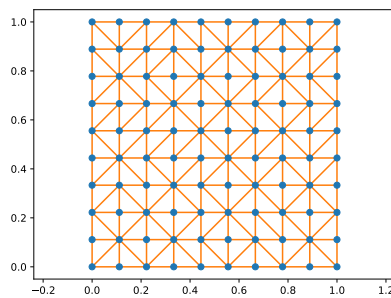
$$u(x, y) = \sum_i c_i \phi_i.$$

Lastni nihajni načini kvadratne opne

Kljub temu, da je naša naloga poiskati lastne nihanje načine opne, si bomo za začetek pogledali lastne nihajne načine kvadratne opne. Na slikah (1) sta prikazani matriki A in B , kjer upoštevamo le notranje elemente triangulacije, prikazane na sliki 2. Matrika A je identična matriki diskretnih diferencialov, ki smo jo srečali v 5. in 6. nalogi. To je slučaj zaradi izbire porazdelitve točk. Obliko matrike B vidimo prvič. Obe matriki sta redki (*sparse*) in simetrični, zato lahko za iskanje lastnih vrednosti uporabimo knjižico `scipy.sparse.linalg.eigsh`.



Slika 1: Globalni matriki za kvadratno opno.

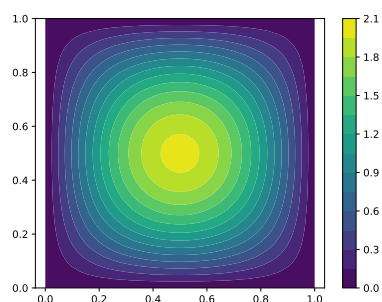


Slika 2: Triangulacija za kvadratno opno za 10x10 točk.

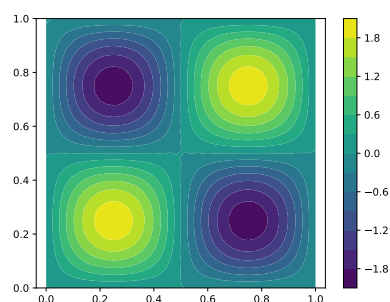
Lastne nihajne načine smo poiskali za 50x50 točk. Lastne vrednosti za prve štiri lastne nihajne načine so prikazane v tabeli 1. Na slikah 3 so nato prikazani lastni nihajni načini pridobljeni s FEM metodo v jeziku *Python*.

n	i j	$\pi^2(i^2 + j^2)$	k^2
1	1 1	19.7392	19.7584
2	1 2	49.3480	49.4512
	2 1		49.4516
3	2 2	78.9568	79.2920
4	3 2	98.6960	99.0636
	2 3		99.0680

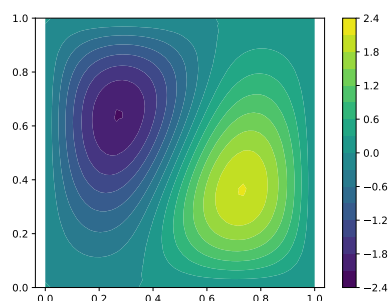
Tabela 1: Lastne vrednosti za prve štiri lastne nihajne načine.



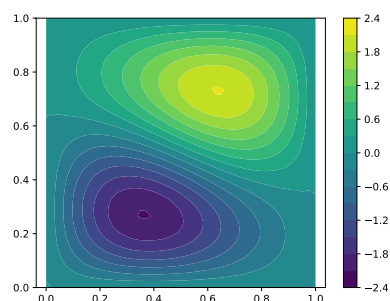
(a) Prvi nihajni način.



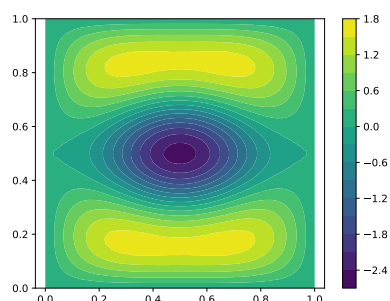
(b) Tretji nihajni način.



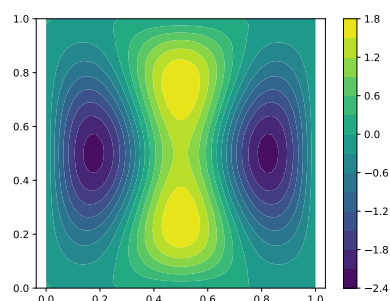
(c) Drugi nihajni način.



(d) Drugi nihajni način.



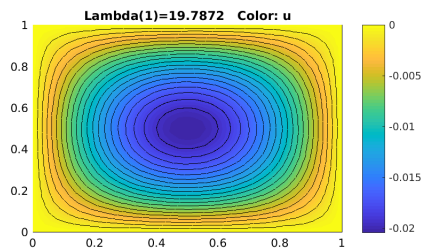
(e) Četrty nihajni način.



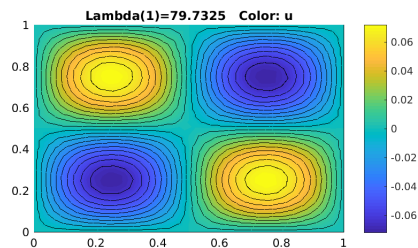
(f) Četrty nihajni način.

Slika 3: Lastni nihajni načini kvadratne opne izračunani v jeziku *Python*.

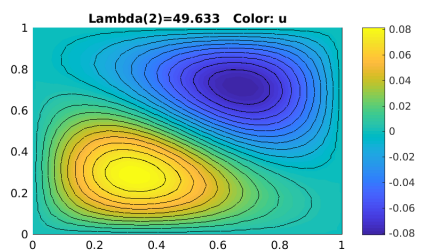
Poračunali pa smo še lastne nihajne načine s pomočjo programa *Matlab*, kateri ima vgrajeno orodje za reševanje parcialnih diferencialnih enačb. Priključimo ga z ukazom `pdetool`. Rešitve so prikazane na slikah 4.



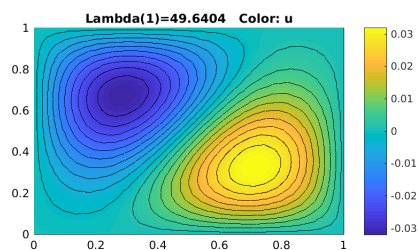
(a) Prvi nihajni način.



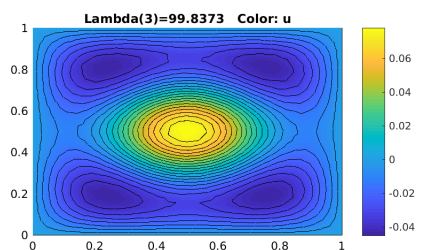
(b) Tretji nihajni način.



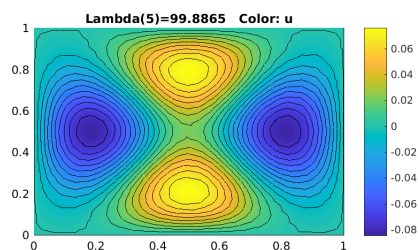
(c) Drugi nihajni način.



(d) Drugi nihajni način.



(e) Četrty nihajni način.

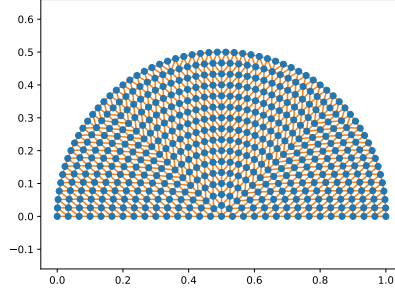


(f) Četrty nihajni način.

Slika 4: Lastni nihajni načini kvadratne opne izračunani v programu *Matlab*.

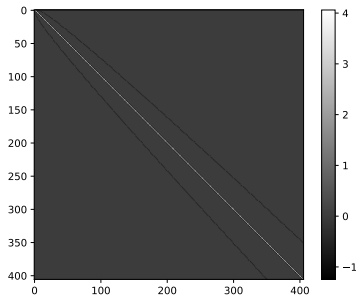
Lastni nihajni načini polkrožne opne

Reševanje je identično kot za primer kvadratne opne, le da imamo tokrat drugačno triangulacijo območja. Triangulacija je prikazana na sliki 5, kjer imamo 406 notranjih točk in 91 robnih točk.

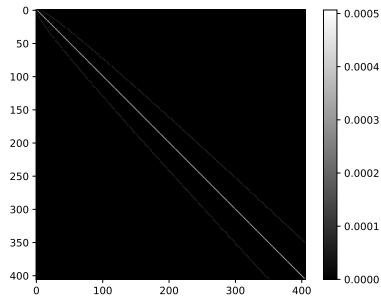


Slika 5: Triangulacija za polkrožno opno za 497 točk.

Ponovno si oglejmo globalni matriki A in B , prikazani na slikah 6. Matriki sta še vedno simetrični in redki (*sparse*), kar ni presenetljivo. Vidimo tudi, da zaradi drugačne porazdelitve točk ne dobimo več matrike A z elementi 4 in -1, temveč se razlikujejo. Izvendiagonalni elementi se širijo narazen. Vzrok tiči v izbiri točk za triangulacijo, namreč na vsakem polkrogu od $r = 0$ do $r = R$ imamo več točk, zato pride tudi do razširjanja izvendiagonalnih elementov. Za izračun lastnih nihajnih načinov ponovno uporabimo knjižico `scipy.sparse.linalg.eigsh`.



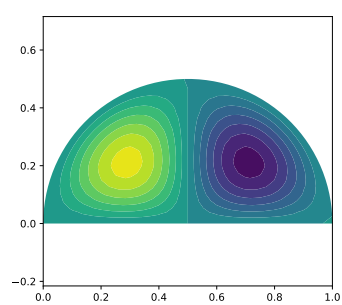
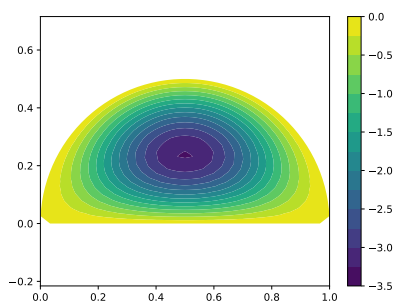
(a) Globalna matrika A .



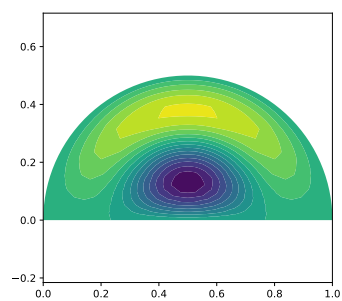
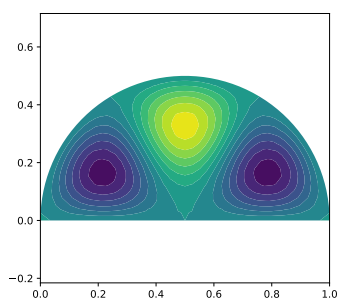
(b) Globalna matrika B .

Slika 6: Globalni matriki za polkrožno opno.

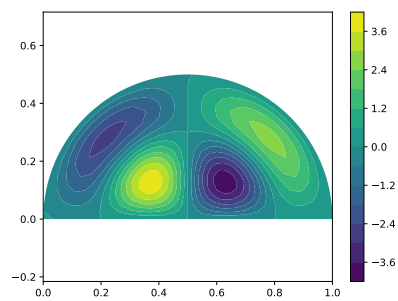
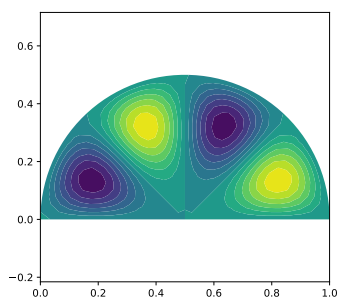
Lastne nihajne načine smo poiskali za 406 notranjih točk. Lastne vrednosti za prvih šest lastnih nihajnih načinov polkrožne opne, so prikazane v tabeli 2. Na slikah 7 so nato prikazani lastni nihajni načini pridobljeni s FEM metodo v jeziku *Python*.



(a) Prvi nihajni način $i = 1, j = 1$. (b) Drugi nihajni način $i = 2, j = 1$.



(c) Tretji nihajni način $i = 3, j = 1$. (d) Četrty nihajni način $i = 1, j = 2$.



(e) Peti nihajni način $i = 4, j = 1$. (f) Šesti nihajni način $i = 2, j = 2$.

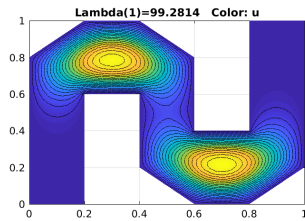
Slika 7: Lastni nihajni načini polkrožne opne izračunani v jeziku *Python*.

n	i j	$\xi_i^j/0.5$	k^2	ξ_i^j
1	1 1	58,7278	59,0190	3,8317
2	2 1	105,4985	106,3977	5,1356
3	3 1	162,8259	164,8956	6,3802
4	1 2	196,8738	200,3545	7,0156
5	4 1	230,3318	234,6404	7,5883
6	2 2	283,4000	290,4187	8,4172

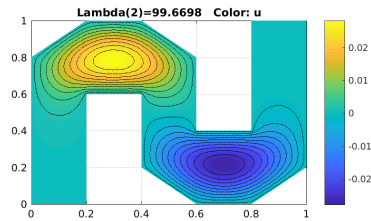
Tabela 2: Lastne vrednosti za prvih 6 lastnih nihajnih načinov polkrožne opne dobljenih v jeziku *Python*.

Lastni nihajni načini danega lika

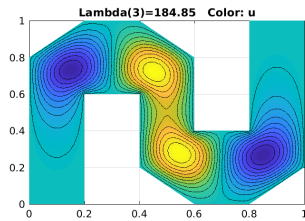
Iskanje lastnih nihajnih načinov je mnogo lažje v orodju *Matlab* v katerem hitreje podamo območje iskanja, triangulacijo pa izvede program sam. Tako smo dobili rezultate, prikazane na slikah 8 in 9.



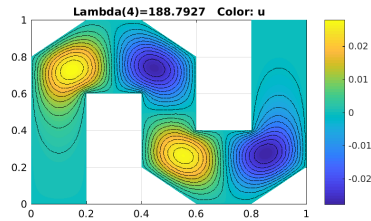
(a) Prvi nihajni način



(b) Drugi nihajni način

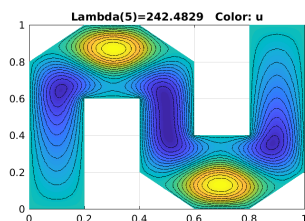


(c) Tretji nihajni način

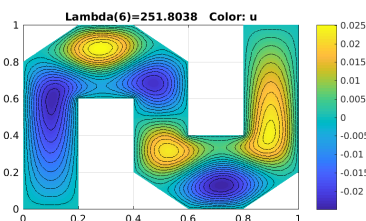


(d) Četrty nihajni način

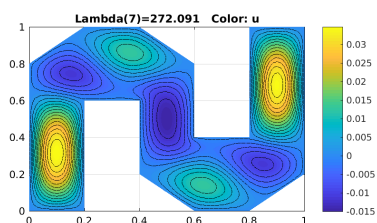
Slika 8: Prvi štirje lastni nihajni načini danega lika



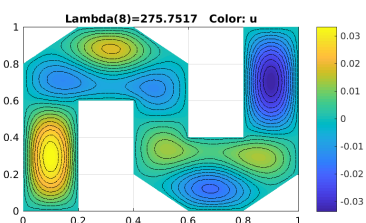
(a) Peti nihajni način



(b) Šesti nihajni način



(c) Sedmi nihajni način



(d) Osmi nihajni način

Slika 9: Peti, šesti, sedmi in osmi lastni nihajni načini danega lika.

Nastavek Galerkinova

Poskusimo oceniti lastne nihajne vrednosti za polkrožno opno z *Galerkinovim* nastavkom

$$\phi_i \rightarrow \phi_{mn}(r, \varphi) = r^{m+n}(1-r) \sin(m\varphi), \quad (6)$$

ki je bazna funkcija napeta čez celoten naš polkrog, pri čemer je

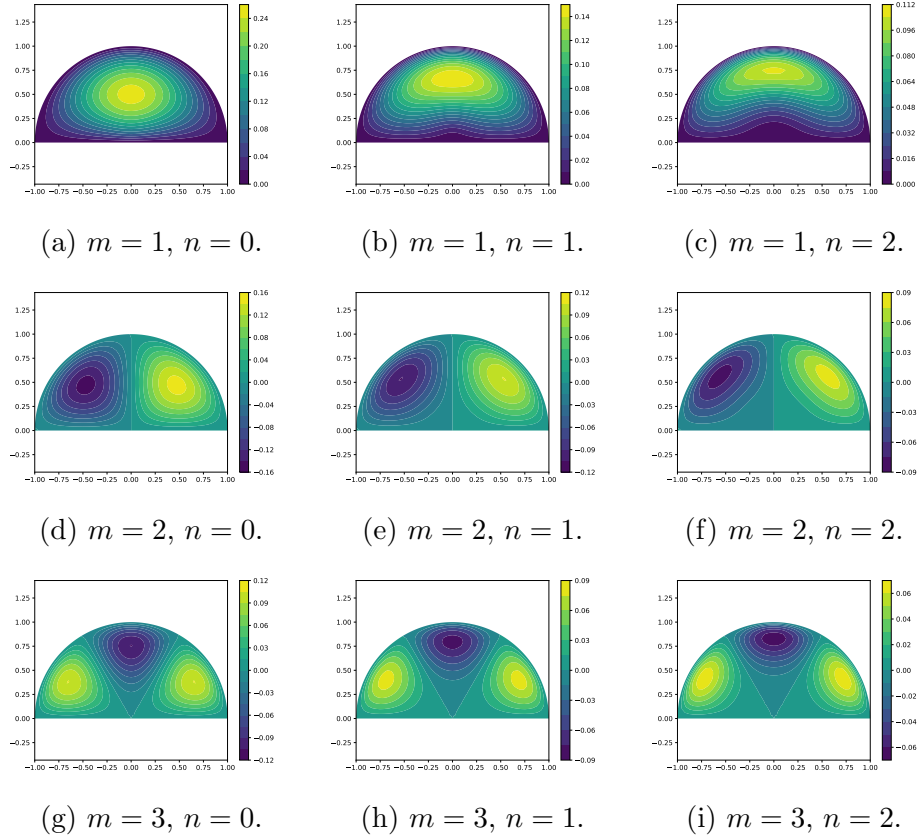
$$1 \leq m \leq m_{max} \quad \text{in} \quad 0 \leq n \leq n_{max}.$$

Za začetek si pogledjmo nekaj baznih funkcij, prikazanih na slikah 10. Določene bazne funkcije spominjajo na lastne nihajne načine. Ideja je tako iz nekaj izbranih baznih funkcij skonstruirati lastne nihajne načine in poiskati lastne vrednosti. Skonstruirati je potrebno matriko A in matriko B ter rešiti sistem

$$A\vec{c} = k^2 B\vec{c}.$$

Tako dobimo rešitev

$$u(x, y) = \sum_i c_i \phi_i.$$



Slika 10: Nekaj prvih baznih funkcij.

Konstrukcija matrik je zelo podobna prejšnji metodi FEM:

$$A_{ij} \rightarrow A_{(mn)(kl)} = \langle \nabla \phi_{mn}, \nabla \phi_{kl} \rangle$$

$$B_{ij} \rightarrow B_{(mn)(kl)} = \langle \phi_{mn}, \phi_{kl} \rangle,$$

tako dobimo bločni matriki. Tokrat moramo je skalarni produkt $\langle a, b \rangle$ definirati drugače, in sicer:

$$\langle a, b \rangle = \int_0^1 r dr \int_0^\pi a \cdot b \, d\varphi.$$

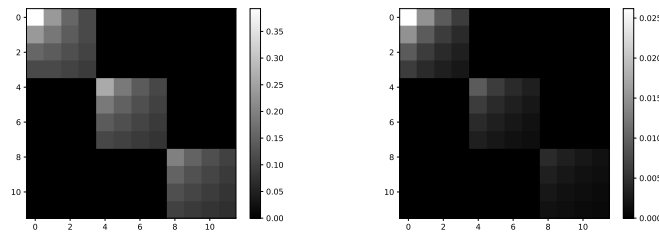
Funkcije $\sin(m\varphi)$ so med seboj ortogonalne in nam olajšajo delo, saj je potrebno poračunati le diagonalne elemente bločnih matrik. Matriki A in B sta torej enaki:

$$A_{(mn),(ml)} = \langle \nabla \phi_{mn}, \nabla \phi_{ml} \rangle$$

$$= \frac{\pi}{2} \left(\frac{nl}{2m+n+l} - \frac{2nl+n+l}{2m+n+l+1} + \frac{(n+1)(l+1)}{2m+n+l+2} \right), \quad (7)$$

$$\begin{aligned}
B_{(mn),(ml)} &= \langle \phi_{mn}, \phi_{ml} \rangle \\
&= \frac{\pi}{2} \left(\frac{1}{2m+n+l+2} - \frac{2}{2m+n+l+3} + \frac{1}{2m+n+l+4} \right).
\end{aligned} \tag{8}$$

Če si pogledamo izračunani matriki na sliki 11, opazimo, da sta simetrični. Zato lahko za pridobitev lastnih vrednosti in vektorjev uporabimo knjižico `scipy.linalg.eigh`.



Slika 11: Matrika A levo in matrika B desno za $m = 3$ in $n = 4$.

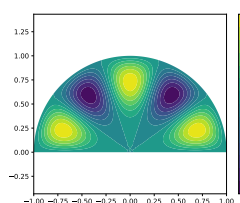
Poračunali smo lastne vrednosti za prvih 13 lastnih vrednosti. Primerjava z analitičnimi je prikazana v tabeli 3.

n	i j	$(\xi_i^j)^2$	k^2	ξ_i^j
1	1 1	14,6819	14,6820	3,8317
2	2 1	26,3746	26,3746	5,1356
3	3 1	40,7065	40,7065	6,3802
4	1 2	49,2185	49,2185	7,0156
5	4 1	57,5829	57,5829	7,5883
6	2 2	70,8500	70,8500	8,4172
7	1 5	76,9389	76,9389	8,7715
8	3 2	95,2776	95,2776	9,7610
9	6 1	98,7263	98,7263	9,9361
10	1 3	103,4995	103,5073	10,1735
11	4 2	122,4278	122,4279	11,0647
12	7 1	122,9076	122,9076	11,0864
13	2 3	135,0207	135,0545	11,6198
14	8 1	149,4529	149,4529	12,2251
15	5 2	152,2412	152,2414	12,3386

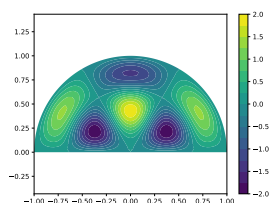
Tabela 3: Nekaj lastnih vrednosti za $m = 10$ in $n = 7$.

Poračunane lastne vrednosti se izjemno ujemajo z analitičnimi. Rezultat je veliko boljši kot pri metodi s trikotniki, tudi izračun lastnih vrednosti in lastnih vektorjev je časovno zelo kratek. Pokazali smo, da lahko z ustreznimi baznimi funkcijami precej hitreje in veliko natančneje dobimo rezultate.

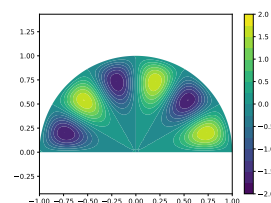
Lastne nihajne načine lahko skonstruiramo iz dobljenih lastnih vektorjev. Poglejmo si le nekaj višjih lastnih nihajnih načinov, prikazanih na slikah 12.



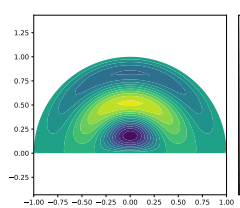
(a) 7 nihajni način.



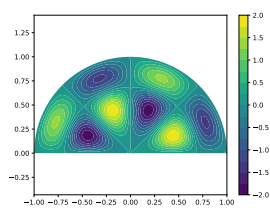
(b) 8 nihajni način.



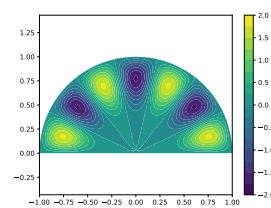
(c) 9 nihajni način.



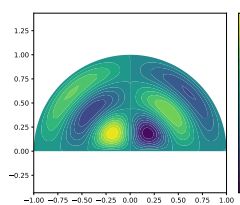
(d) 10 nihajni način.



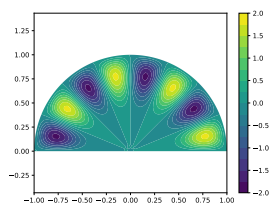
(e) 11 nihajni način.



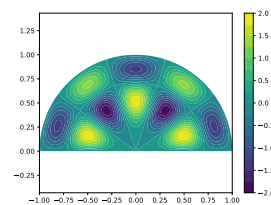
(f) 12 nihajni način.



(g) 13 nihajni način.



(h) 14 nihajni način.



(i) 15 nihajni način.

Slika 12: Nekaj lastnih nihajnih načinov.