

University of *Ljubljana*
Faculty of *Mathematics and Physics*



Modelska Analiza 2

6. naloga - Parcialne diferencialne enačbe: lastne rešitve

Avtor: Matic Lubej
Asistent: dr. Simon Čopar
Predavatelj: prof. dr. Alojz Kodre

Ljubljana, 2.4.2014

Naloga:

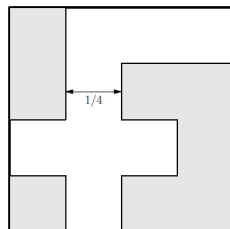
Pri tej nalogi smo se ponovno ukvarjali z numeričnim reševanjem parcialnih diferencialnih enačb z robnimi problemi. Tokrat smo enačbe reševali z lastnimi funkcijami. V prvem delu naloge smo imeli kvadratno opno, ki smo jo lahko enakomerno obtežili, ali pa obtežili neenakomerno v obliki črke „f“. V drugem delu naloge smo enako storili še za polkrožno cev, vendar le za primer enakomerno obtežene polkrožne opne.

Del I

Nihanje kvadratne opne

1 Naloga

Določi nekaj najnižjih lastnih frekvenc in lastnih nihajnih načinov kvadratne opne, ki je odebeljena na večjo ploskovno gostoto povsod, razen na območju v obliki črke **f**, kakor je prikazano na skici. Prevedi problem na sistem diferenčnih enačb. Kako se spremeni situacija, če je senčeno območje lažje? Konvergenco metode preizkusi na neobteženi opni.



2 Uvod

Imamo valovno enačbo v ravnini, ki jo zapišemo kot:

$$\nabla^2 u = \frac{1}{c(x, y)^2} \frac{\partial^2 u}{\partial t^2}, \quad c(x, y)^2 = \frac{\gamma}{\sigma(x, y)}, \quad (1)$$

kjer je c hitrost valovanja v točki (x, y) , γ je površinska napetost opne, σ pa ploskovna gostota opne. Uporabimo lahko separacijo spremenljivk in uporabimo nastavek:

$$u(x, y, t) = u_0(x, y)e^{-i\omega t} \rightarrow u(x, y)e^{-i\omega t}. \quad (2)$$

Naša enačba se nekoliko poenostavi, dobimo:

$$\nabla^2 u = \frac{\omega^2 \sigma}{\gamma} u = \frac{\omega^2}{c^2} u = k^2 u. \quad (3)$$

Spet torej numerično rešujemo parcialno diferencialno enačbo, vendar tokrat z metodo lastnih rešitev. Enačbo (3) tudi tokrat diskretiziramo:

$$\nabla^2 u \rightarrow \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}}{h^2}$$

in če operator ∇^2 skonstruiramo v matriko, matriko $u_{i,j}$ pa v vektor, lahko problem zapišemo kot matrično enačbo:

$$\mathbf{A}u = \mathbf{B}u, \quad (4)$$

kjer je matrika \mathbf{A} operator ∇^2 v diferenčni obliki, operator \mathbf{B} pa diagonalna matrika vrednosti $\frac{\omega^2 \sigma_{i,j}}{\gamma}$, ki vsebuje informacijo o strukturi opne na določenem koščku (i, j) . Če ima opna enakomerno ploskovno gostoto, potem je \mathbf{B} kar identiteta, če pa ni, pa imamo lahko težave, ker $\mathbf{B}^{-1}\mathbf{A}$ ni več simetrična matrika. To rešimo tako, da za $\mathbf{B} = \text{diag}(\alpha_1, \dots, \alpha_N)$ transformiramo:

$$A_{i,j} \rightarrow \frac{A_{i,j}}{\sqrt{\alpha_i \alpha_j}}. \quad (5)$$

Na tak način upoštevamo različno ploskovno gostoto, hkrati pa ohranimo matriko \mathbf{A} simetrično. Na matriki \mathbf{A} nato uporabimo splošno znane metode za diagonalizacijo, da dobimo lastne vrednosti in lastne vektorje, kjer so nekateri nihajni načini degenerirani. Če lastne vektorje nazaj zapišemo kot matriko, dobimo lastne nihajne načine, ustrezne lastne vrednosti pa predstavljajo lastne frekvence določenega lastnega nihanja. Na tak način dobimo set lastnih funkcij in lastnih vrednosti, torej lahko kakršnokoli funkcijo razvijemo po teh lastnih rešitvah in določimo tudi časovno odvisnost taknega nihanja. Če imamo začetni odmik $u_0(x, y)$, ki je v splošnem neka kompleksna funkcija, določimo koeficiente $C_{m,s}$, ki predstavljajo amplitudo prispevka ortonormirane lastne funkcije v rešitvi, kot:

$$C_{m,s} = \int_S u_0(x, y) A_{m,s}(x, y) dS = h^2 \sum_{i,j} u_{0,(i,j)} A_{m,s,(i,j)}. \quad (6)$$

Časovno odvisna rešitev je tako vsota prispevkov lastnih funkcij, kar zapišemo kot:

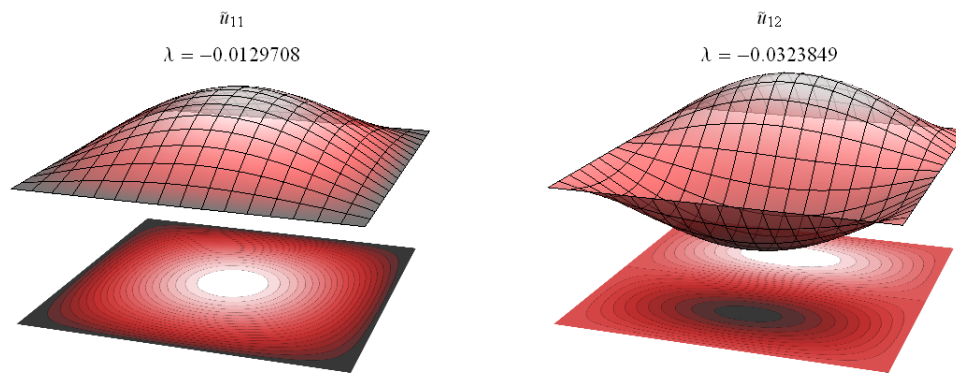
$$u(x, y, t) = \sum_{m,s} C_{m,s} A_{m,s} e^{i\omega_{m,s} t}. \quad (7)$$

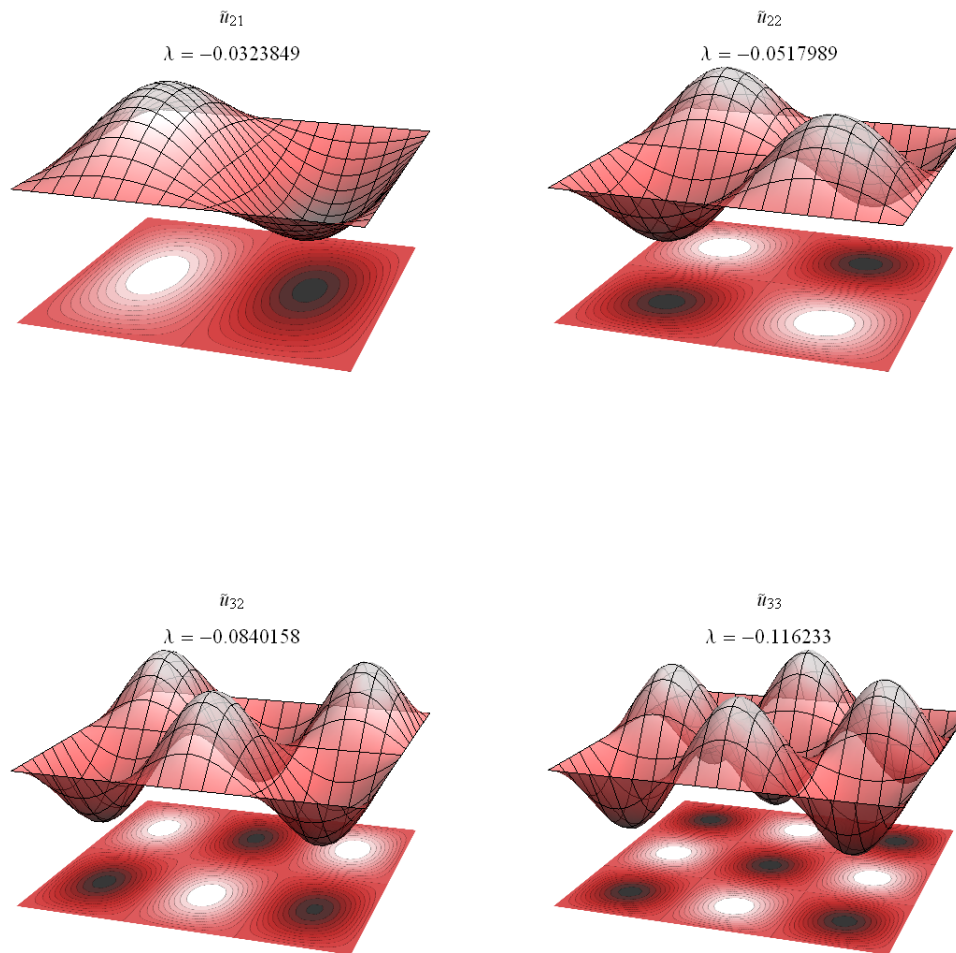
3 Reševanje

Reševanja sem se tokrat lotil v C++ s pomočjo knjižnic iz Numerical Recipes. Skonstruiral sem matriko \mathbf{A} , ki sem jo nato dal v rutini `tred2` in `tqli`, ki najdeta ustrezno tridiagonalno matriko matriki \mathbf{A} , nato pa jo diagonalizirata in vrneta lastne vrednosti ter funkcije. Te sem nato izvozil v program Mathematica, kjer sem rešitve tudi narisal. Pri tej nalogi se je izkazalo, da je diagonalizacija bolj zahtevna od relaksacije iz prejšnje naloge, zato sem se zadovoljil z razdelitvijo mreže 50×50 v smeri x in y iz intervala $(x, y) \in [0, 1] \times [0, 1]$. Kasneje sem isto kodo preizkusil še v Mathematici, ki se je izkazala za bolj optimizirano, saj je isti problem enako dobro rešila v veliko hitrejšem času.

4 Rezultati

Za začetek si oglejmo lastna nihanja kvadratne opne v primeru enakomerno obtežene opne. Ta del naloge služi kot test pravilnosti. Oglejmo si nekaj prvih lastnih nihajnih načinov:





Vidimo, da so opisane funkcije res pravilne lastne funkcije, ki jih opisuje enačba

$$f_{m,n}(x,y) = \sin(\pi x n/a) \sin(\pi y m/b), \quad (8)$$

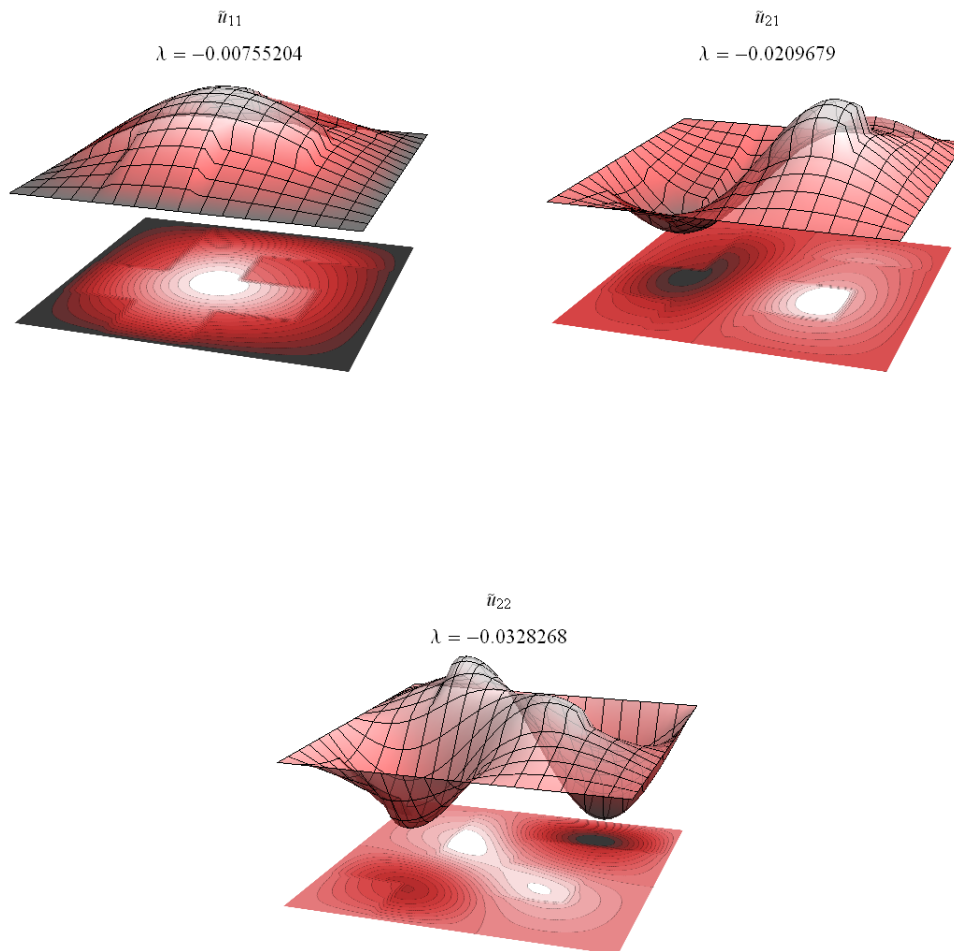
kjer m in n predstavljata indeksa (m,n) -to lastno funkcijo, a in b pa velikosti stranic. Za lažjo predstavo prilagam tudi povezavo do animacije:

https://dl.dropboxusercontent.com/u/12747812/Animacije/1_anim_mod6.gif

Oglejmo si še animacijo, kjer sem poljubno začetno stanje za enakomerno obteženo opno razvil po orto-normirani bazi:

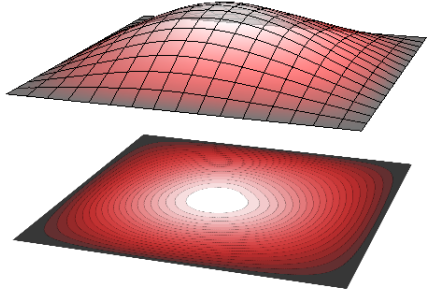
https://dl.dropboxusercontent.com/u/12747812/Animacije/1_eq.gif

Oglejmo si sedaj rešitve, kjer uporabimo neenakomerno obteženo opno. V tem primeru sem uporabil utež $w_f = 1$ v notranjosti črke „f“ in $w_o = 2$ po zunanem območju. Najprej si oglejmo lastne rešitve matrike \mathbf{A} iz enačbe (5), ki niso lastne rešitve našega sistema, ampak rešitve simetrizirane matrike \mathbf{A} :

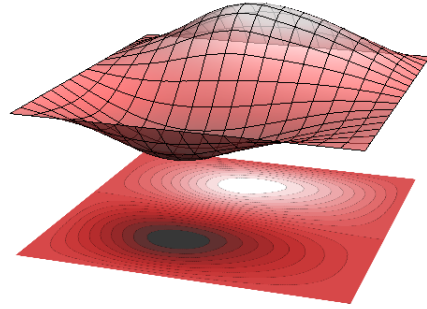


Na vseh rešitvah je tudi opazna črka „f“. Če si ogleđamo lastne funkcije u , ki smo jih transformirali nazaj kot $u_j \rightarrow u_j / \sqrt{\alpha_j}$, potem dobimo bolj homogene rešitve, ki pa med sabo niso več ortogonalne:

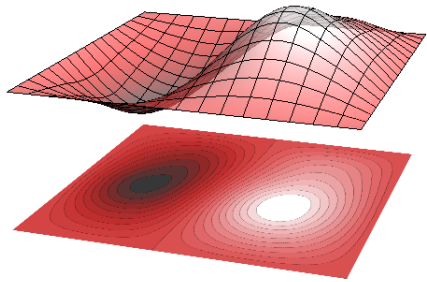
\bar{u}_{11}
 $\lambda = -0.00755204$



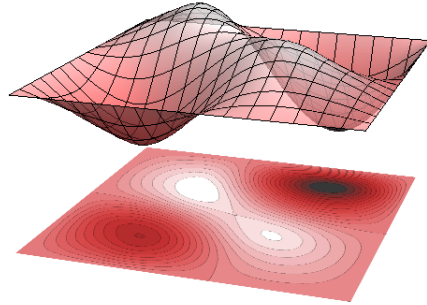
\bar{u}_{12}
 $\lambda = -0.019037$

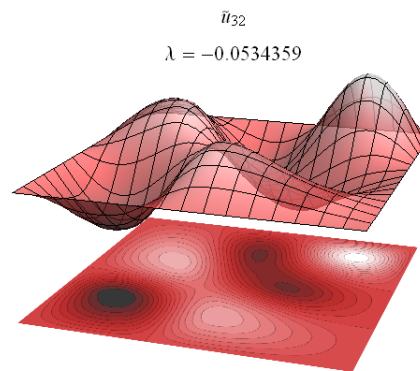
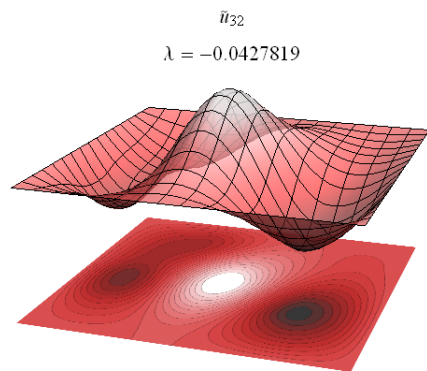


\bar{u}_{21}
 $\lambda = -0.0209679$



\bar{u}_{22}
 $\lambda = -0.0328268$

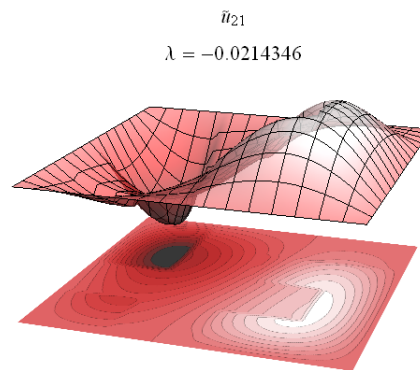
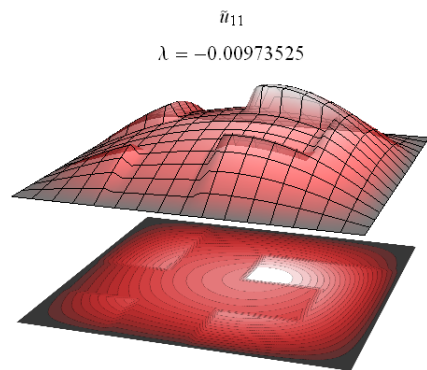


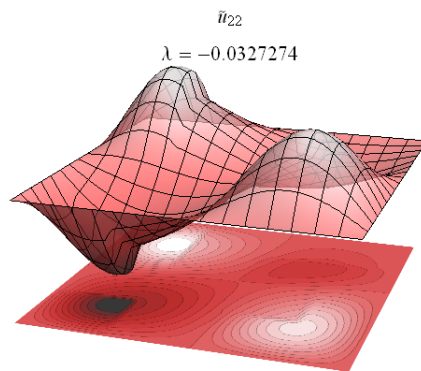


Rešitve so, v primerjavi s prejšnjimi, nekoliko deformirane, kar je smiselno. Deformacije se pojavljajo v skladu s težiščem opne. Opazimo enak trend med različnimi lastnimi funkcijami. Tudi v tem primeru si lahko ogledamo animacijo, za poljubno začetno stanje:

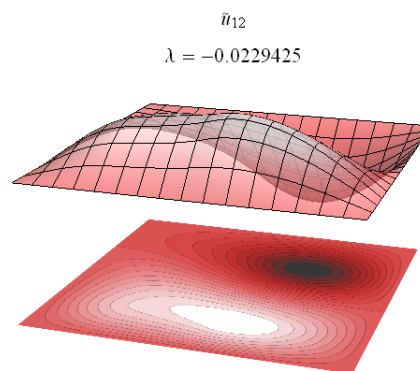
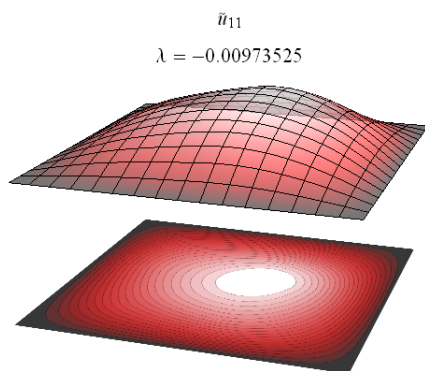
https://dl.dropboxusercontent.com/u/12747812/Animacije/1_he.gif

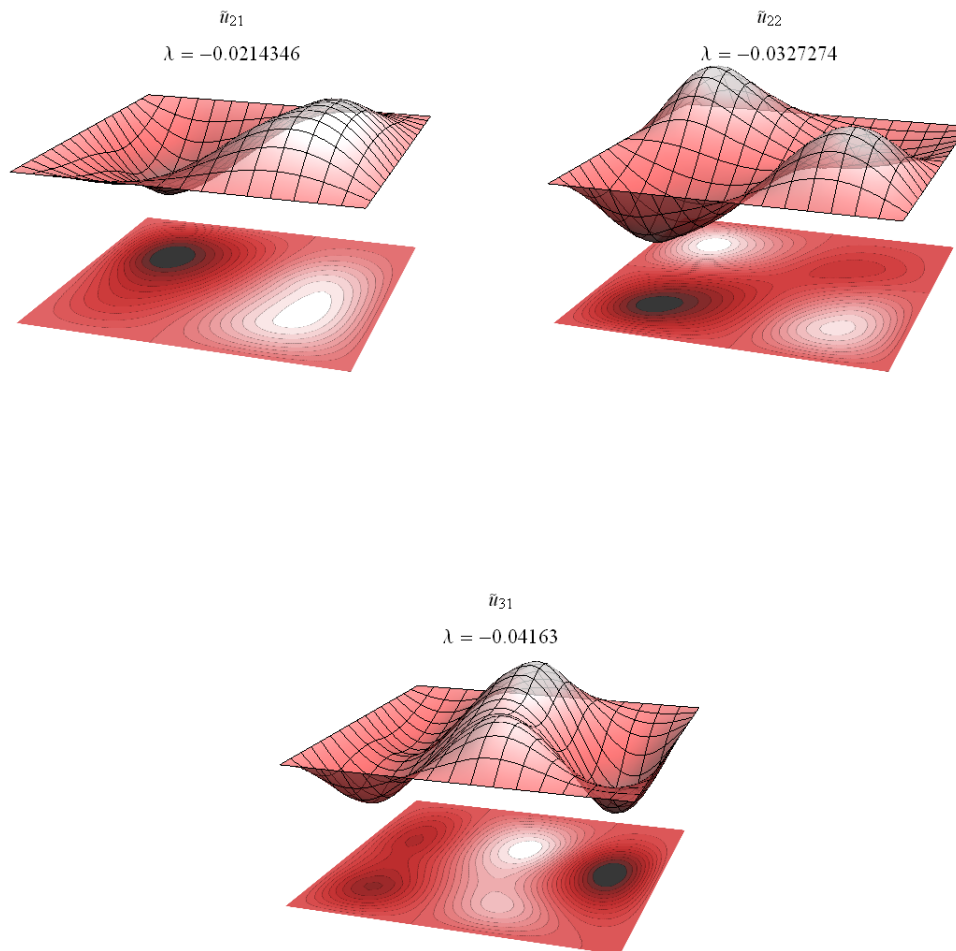
Kaj pa če obratno obtežimo opno? V tem primeru sem uporabil utež $w_f = 1.25$ v notranjosti črke „f“ in $w_o = 0.75$ po zunanem območju. Najprej si oglejmo lastne rešitve matrike \mathbf{A} iz enačbe (5), ki niso lastne rešitve našega sistema, ampak rešitve simetrizirane matrike \mathbf{A} :





Na vseh rešitvah je tudi opazna črka „f“. Če si ogledamo lastne funkcije u , ki smo jih transformirali nazaj kot $u_j \rightarrow u_j/\sqrt{\alpha_j}$, potem dobimo bolj homogene rešitve, ki pa med sabo niso več ortogonalne:



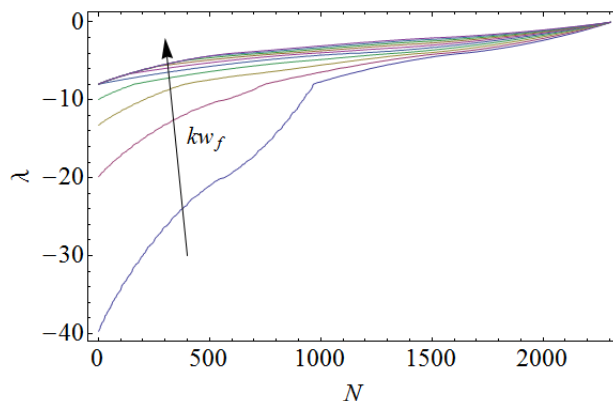


Čeprav ni tako očitno, se lastne rešitve v tem primeru razlikujejo od prejšnjih. Tudi v tem primeru si lahko ogledamo animacijo, za poljubno začetno stanje:

https://dl.dropboxusercontent.com/u/12747812/Animacije/1_li.gif

5 Lastne vrednosti

Ogledamo si lahko še lastne vrednosti za različna razmerja w_f in w_o . Spodaj so prikazani rezultati za $w_f = 1$ in $w_o = kw_f$, kjer sem spreminjal vrednost parametra k :



Vidimo, da najnižje lastne vrednosti začnejo v isti vrednosti, potem pa se razcepijo.

Del II

Nihanje polkrožne opne

1 Naloga

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

2 Uvod in reševanje

V tem delu naloge sem mislil, da se bolj splača, če problem predstavimo v polarnih koordinatah na pravokotniku $(r, \varphi) \in [0, 1] \times [0, \pi]$, kjer sem uporabil enako število točk v obeh koordinatah, torej se širini korakov v obeh koordinatah razlikujeta, kar moramo upoštevati v naših enačbah. Operator ∇^2 v polarnih koordinatah razpišemo kot:

$$\nabla^2 = \frac{1}{r} \frac{\partial}{\partial r} + \frac{\partial^2}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2}{\partial \varphi^2}, \quad (9)$$

v diferenčni obliki pa je to enako:

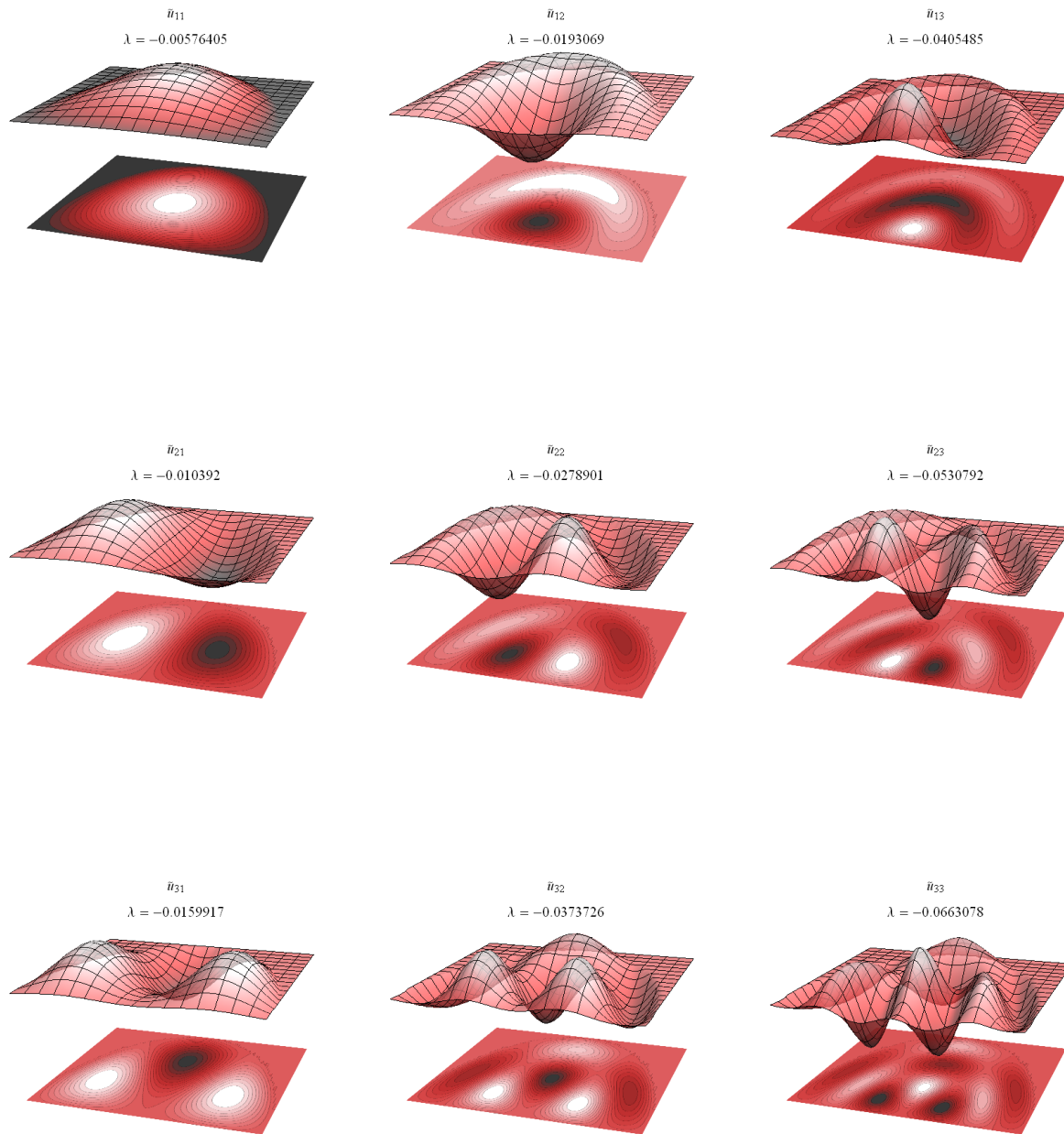
$$h^2 \nabla^2 = u_{i-1,j} \left(1 - \frac{h}{2r_i}\right) + u_{i+1,j} \left(1 + \frac{h}{2r_i}\right) + u_{i,j-1} \frac{1}{\pi^2 r_i^2} + u_{i,j+1} \frac{1}{\pi^2 r_i^2} - 2u_{i,j} \left(1 + \frac{1}{\pi^2 r_i^2}\right). \quad (10)$$

Izkazalo se je, da sem imel s takšnim reševanjem kar nekaj težav, na koncu pa sem zaradi veliko porabljenega časa obupal. Odločil sem se, da preizkusim problem rešiti še v kartezičnih koordinatah, kjer s tem žrtvujemo oster rob, kar pa pri velikem številu točk ne bi smel biti problem. V tem primeru sem problem reševal na $2n \times n$ velikem polju. Za majhno vrednost n sem si zapisal enačbe in matriko nato razširil za splošno vrednost n na podlagi vzorcev v matriki. Dobil sem zadovoljive rezultate. Prilagam povezavo do animacije lastnih nihajnih načinov za ta primer:

https://dl.dropboxusercontent.com/u/12747812/Animacije/2_anim_mod6.gif

3 Rezultati

Oglejmo si lastne nihajne načine polkrožnega polja:



Tudi tokrat funkcije izgledajo pravilne oblike, kot nam narekuje analitična formula za lastne rešitve polkroga:

$$f_{m,n}(x, \varphi) = J_{2m+1}(y_{m,s}x) \sin(\varphi(2m+1)), \quad (11)$$

kjer $J_{2m+1}(x)$ predstavlja $(2m+1)$ -to Besselovo funkcijo, $y_{m,s}$ pa s -to ničlo m -te Besselove funkcije.