SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Marko Dominković

RAČUNANJE REZONANTNIH MODOVA U SUSATAVU BEM++

Diplomski rad

Voditelj rada:
Luka Grubišić

Zagreb, studenoga, 2018

Ovaj diplomski rad obranjen je dana ____________________ pred ispitnim povjerenstvom u sastavu:

1. ______________________________, predsjednik
2. ______________________________, član
3. ______________________________, član

Povjerenstvo je rad ocijenilo ocjenom ________________.

Potpisi članova povjerenstva:

1. ______________________________
2. ______________________________
3. ______________________________

Samom sebi

Sadržaj

Sadržaj **iv**

Uvod **1**

1 Rezonance i modovi **3**

2 BEM++ **5**
2.1 O BEM++ 5
2.2 Primjena BEM++ 5

3 **9**
3.1 Računanje najmanje svojstvene vrijednosti 9

4 Naslov poglavlja u sadržaju **13**
4.1 Naslov sekcije u sadržaju 13

Bibliografija **17**

Uvod

U ovom radu ćemo implementirat metodu za rješavanje problema računanja rezonatnih modova u problemu akustičkog ili elektromagnetskog raspršenja. Za rješavanje danih problema koristit ćemo BEM++ sustav.

Poglavlje 1

Rezonance i modovi

Sustav

$$(I + A(w))f_0 = 0 \tag{1.1}$$

ima netrijvijalno rješenje za f_0 samo ako operator $(I + A(w))^{-1}$ je singularan za određenu frekvenciju w.

Poglavlje 2

BEM++

2.1 O BEM++

[2] Bem++ je sustav za rješavanje akustičnih, elektrostatičkih i elektromagnetnih problema pomoću metode rubnih elemenata (boundary element method). Moguće ga je koristiti kroz sučelja programskih jezika Python i C++.

2.2 Primjena BEM++

Rešetke i funkcijski prostori

Za prikaz oblika nekog tijela u BEM++ koristimo se rešetkama. Rešetka je skup točaka u prostoru (dvodimenzionalnom ili trodimenzionalnom) i skup bridova koji povezuju... Konstruirajmo jednu jednostavnu rešetku

```
import bempp.api
import numpy as np

grid = bempp.api.shapes.regular_sphere(3)
```

Drugi način konstruiranja rešetke je pomoću polja vrhova i elemenata koji opisuju kako su vrhovi povezani. Sa `vertices` označimo polje svih vrhova. `evaluation_points` opisuje način na koji su ovrhovi povezani.

```
vertices = np.array([[0,1,2,1],
                     [0,2,0,1],
```

```
                    [0,0,0,2]])

evaluation_points = np.array([[0,0,0,1],
                              [1,1,2,2],
                              [2,3,3,3]])

grid = bempp.api.grid_from_element_data(vertice, evaluation_points)
```

Prvi element povezuje vrhove 0, 1 i 2. Drugi povezuje 0, 1 i 3 itd. Rešetke možemo učitati iz .msh datoteka. BEM++ podržava Gmsh v2.2 msh format

```
grid = bempp.api.import_grid('my_grid.msh')
```

Važnu ulogu u BEM++ imaju funkcijski prostori. Za inicijalizaciju funkcijskog prostora potrebna nam je rešetka.

```
space = bempp.api.function_space(grid, "P", 0)
```

Prvi argument funkcije je rešetka. Drugi argument je tip prosotra, u ovom slučaju `"P"` označava prostor polinoma.

Operatori

Operator A difiniramo kao

$$A : \mathcal{D} \to \mathcal{R}$$

preslikavanje s domene D na kodomenu R, gdje su D i R definirane na površini dane rešetke. BEM++ ne koristi s direktno rubnim operatorom A, već slabijom formom

$$a(u, v) := \int_\Gamma [Au](\mathbf{y})\overline{v(\mathbf{y})}d\mathbf{y}, \quad u \in \mathcal{D}, v \in \mathcal{V} \tag{2.1}$$

gdje je $\mathcal{V}$ dualni prostor prostora $\mathcal{R}$. Rubni operatori se nalaze u paketu `bempp.api.operators.boundary`. Neka je $g(x, y)$ Greenova funkcija. Definiramo jednoslojni $\mathcal{V}$ i dvoslojni $\mathcal{K}$ potencijalni operator sa:

$$[\mathcal{V}\Phi](\mathbf{x}) = \int_\Gamma g(\mathbf{x}, \mathbf{y})\Phi(\mathbf{y})ds(\mathbf{y}), \quad \mathbf{x} \in \mathbb{R}^3 \backslash \Gamma \tag{2.2}$$

$$[\mathcal{K}\Phi](\mathbf{x}) = \int_\Gamma \frac{\partial g(\mathbf{x}, \mathbf{y})}{\partial \nu(\mathbf{y})} g(\mathbf{x}, \mathbf{y})\Phi(\mathbf{y})ds(\mathbf{y}), \quad \mathbf{x} \in \mathbb{R}^3 \backslash \Gamma \tag{2.3}$$

Sada iz njih izvodimo sljedeće granične operatore:

Operator	Formula
Jednoslojni rubni operator	$[V\phi](x) = \int_\Gamma g(\mathbf{x},\mathbf{y})\phi(\mathbf{y})d\mathbf{y}, \quad \mathbf{x}\in\Gamma$
Dvoslojni rubni operator	$[K\phi](\mathbf{x}) = \int_\Gamma \frac{\partial g(\mathbf{x},\mathbf{y})}{\partial\nu(\mathbf{y})}\phi(\mathbf{y})d\mathbf{y}, \quad \mathbf{x}\in\Gamma$
Adjungirani dvoslojni rubni operator	$[K^{'}\phi](\mathbf{x}) = \int_\Gamma \frac{\partial g(\mathbf{x},\mathbf{y})}{\partial\nu(\mathbf{y})}\phi(\mathbf{y})d\mathbf{y}, \quad \mathbf{x}\in\Gamma$
Hipersingularni rubni operatori	$[D\phi](x) = -\frac{\partial}{\partial\nu(\mathbf{x})}\int_\Gamma \frac{\partial g(\mathbf{x},\mathbf{y})}{\partial\nu(\mathbf{y})}\phi(\mathbf{y})d\mathbf{y}, \quad \mathbf{x}\in\Gamma$

Dostupni operator s obzirom na odabir Greenove funkcije su:

PDJ	Greenova funkcija	Modul
Laplace $-\Delta u = 0$	$g(x,y) = \frac{1}{4\pi\vert x-y\vert}$	`bempp.api.operators.boundary,laplace`
Helmholtz $\Delta u + k^2 u = 0$	$g(x,y) = \frac{e^{ik\vert x-y\vert}}{4\pi\vert x-y\vert}$	`bempp.api.operators.boundary.helmholtz`
Modified Helmholtz $-\Delta u + w^2 u = 0$	$g(x,y) = \frac{e^{-w\vert x-y\vert}}{4\pi\vert x-y\vert}$	`bempp.api.operators.boundary.modified_helmholtz`

Svi rubni operatori su definirani sa domenom, kodomenom i dualni prostorom kodomene. U BEM++ se implementiraju na sljedeći naći:

```
grid = bempp.api.shapes.regular_sphere(3)
space = bempp.api.function_space(grid, "DP", 0)
slp = bempp.api.operators.boundary.laplace.
      single_layer(space, space, space)
```

Rješavanje linearnih jednadžbi

Promotrimo sljedeći primjer

$$A\phi = f$$

gdje su f i ϕ funkcijske mreže. Zadanu jednadžbu možemo zapisati u njezinoj slabijoj formi:

$$< Au, v >=< f, v >$$

za sve v u dualnom prostoru. Uobičajni način rješavanje ovakvog sustava u Bempp je sljedeći:

1. Izračunati slabu formu od A pomoću

```
A_discrete = A.weak_form()
```

2. Računanje projekcije f na dualni prostor

```
p = f.projections(A.)
```

3. Pomoću funkcije `gmres`, iz SciPy paketa, riješimo sustav

```
x, info = gmres(discrete_op, p)
```

4. Iz rezultata `x` konstruirajmo funkcijsku mrežu sa

```
phi = bempp.api.GridFunction(A.domain, coefficients=x)
```

Poglavlje 3

3.1 Računanje najmanje svojstvene vrijednosti

Neka je $A \in M_n$ kvadratna matrica reda n, $n \in \mathbb{N}$. Najmanja svojstvena vrijednost matrice A ujedno je i najveća svojstvena vrijednost matrice A^{-1}. Najveća svojstvena vrijednost može se izračunati iterativnom metodom. Opišimo iterativnu metodu za dani matricu A. Za početak odaberimo vektor $b_0 \in \mathbb{R}^n$. Definirajmo niz (b_n) na sljedeći način:

$$b_{k+1} = \frac{Ab_k}{||Ab_k||}, \quad k \in \mathbb{N}$$

Navedeni niz konvergira prema svojstvenom vektoru $x \in \mathbb{R}^n$ čija je odogovarajuća svojstvena vrijednost λ najveća svojstvena vrijednost matrice A. Za računanje najmanje svojstvene vrijednosti treba izračunati inverz matrice A, A^{-1}, s pretpostavkom da je matrica A regularna. Pritom primjenimo iterativnu metodu sa A^{-1}. Naravno, time se složenost samog algortima povećava. Druga mogučnost je rješavanje jednadžbe

$$Ax = b \tag{3.1}$$

u svakoj od iteraciji, gdje je $A \in M_n$, a $b \in \mathbb{R}^n$. Broj iteracija ovisi o izboru početnog vektora i točnosti koju želimo postići. Svojstvena vrijednost jednaka je inverzu norme svojstvenog vektora, $\lambda = 1/||x||$. Opišimo algoritme za računanje najmanje singularne vrijednosti matrice A.

Prvi aloritam je modificirani oblik iterativne metode u kojem imamo z

Algorithm 1: Najmanja singularna vrijednost

Ulaz: Regularna kvadratna matrica A veličine $n \times n$
Izlaz: Najmanja singularna vrijednost

```
b = slučajni vektor ;
e = 0.0001 (točnost);
razlika = 1;
while razlika ≥ e do
    riješi jednandžbu Ax = b;
    x = x/||x||;
    razlika = ||b − x||;
    b = x;
end
return 1/||x||
```

U drugom algoritmu najprije izračunamo inverz matrice A pa potom provodimo klasični iterativi algoritam za traženje najveće svojstvene vrijednosti.

Algorithm 2: Najmanja singularna vrijednost: inverz matrice

Ulaz: Regularna kvadratna matrica A veličine $n \times n$
Izlaz: Najmanja singularna vrijednost

```
b = slučajni vektor, e = 0.0001 (točnost), razlika = 1;
A_inverz = A^{-1} ;
while razlika ≥ e do
    x = A_inverz b;
    x = x/||x||;
    razlika = ||b − x||;
    b = x;
end
return 1/||x||
```

Koji od navedenih algoritama je bolji? Naime, složenost rješavanja sustava $Ax = b$, gdje je A matrica veličine $n \times n$, je $O(n^3)$. Isto tako, računanje inverza matrice A je složenosti $O(n^3)$. Prednost ovog načina je ta da se inverz matrice izračuna samo jednom u odnosu na prvi alogritam, gdje se prilikom svake od iteracije rješava sustav 3.1. Broj iteracija je ve

Poglavlje 4

Naslov poglavlja

4.1 Naslov sekcije

Naslov podsekcije

Teorem 4.1.1. *Iskaz teorema u kojem se javljaju skupovi* $\mathbb{N}$*,* $\mathbb{Z}$*,* $\mathbb{Q}$*,* $\mathbb{R}$ *i* $\mathbb{C}$*.*

Slutnja 4.1.2. *Iskaz slutnje u kojoj se javljaju funkcije* tg*,* th *i* sh*.*

Korolar 4.1.3. *Iskaz posljedice u kojoj se javljaju skupovi* $\operatorname{Ker} T$ *i* $\operatorname{Im} T$*..*

Dokaz. Dokaz posljedice se nalazi u [1]. Pogledajte i [3], [4] te [2]. □

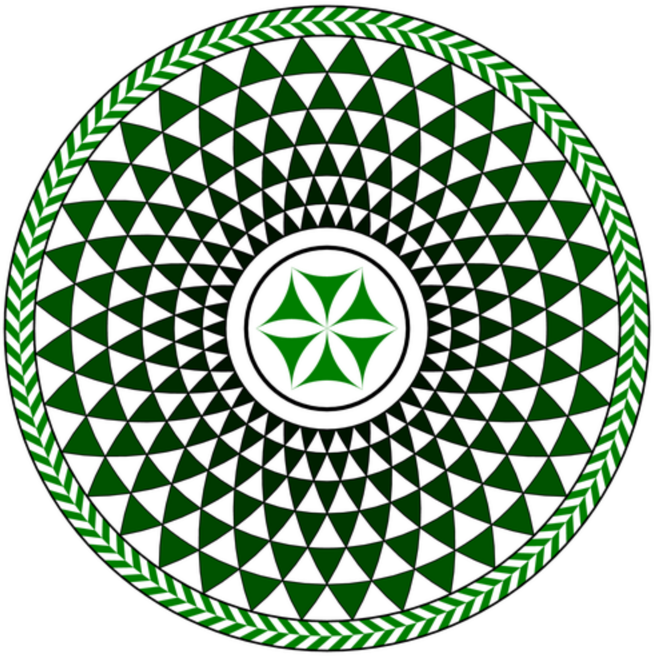
jsfdsqF SG SFG FSG DF GS FG SFG SFG
SFG
SG SDFG SF GS
DG SD S
SD
DFGSDFG
SDGSDF
SDGSDGF
SDGFSFDG
SDGSDG sdfsfg f fdh fgj gh jgjk jkj k yk k klk l fs fd gsdfg dfh dfghj fj ghjk gjk jlk sdf $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ $x_1 + x_2 + x_3 + x_4 = 12$ SDGSG sdfsfg f fdh fgj gh jgjk jkj k yk k klk l fs fd gsdfg dfh dfghj fj ghjk gjk jlk sdf sfdh j fj tuk ugad h j yrtu iru i

$$z\left(1 + \sqrt{\omega_{i+1} + \zeta - \frac{x+1}{\Theta+1} y + 1}\right) = 1$$

GSDFGSDFG

$$1 + 1 = 2 \tag{4.1}$$

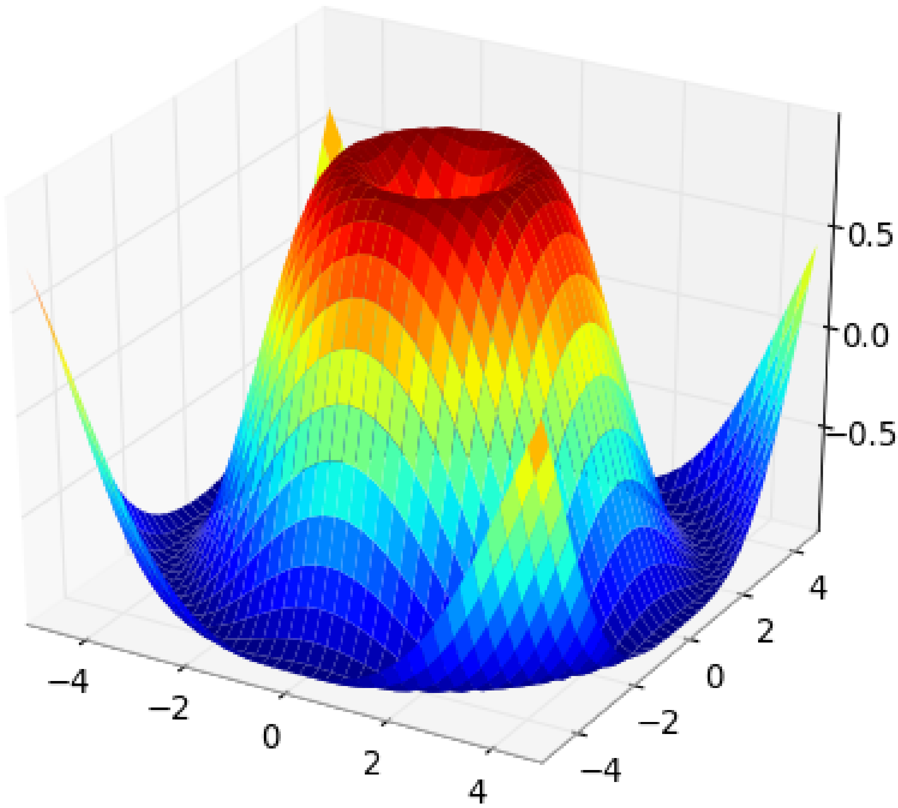
dasdsa SDFGS
SDFGSFG
SFGSFG
SDFGSFG
SDGSFG
Na stranici 14 se nalaza slika u **png** formatu.



Slika 4.1: Prva slika

Na slici 4.2 se nalazi 3D graf neke funkcije.
kao i jedna vrlo komplicirana formula koja slijedi iz (4.1)

$$\sum_{i=1}^{\infty} A_{x_1} \times A_{\alpha_2} \oslash \iint_{\Omega} x^2 \ddagger \limsup_{n\in\mathbb{N}} \frac{\alpha+\theta+\gamma}{n^{\omega}} \text{ je u stvari } \biguplus_{r\in\mathbb{Q}} \overline{\Xi_i \underset{\substack{j\in\mathbb{C}\\ j\ni i\mathbb{Q}}}{\Theta} \Upsilon^{k^j} \underset{*}{\Psi} \hbar|_{\{\alpha\}}}.$$



Slika 4.2: Druga slika

Bibliografija

[1] I. Autor, *Naslov Knjige*, Samizdat, 2052.

[2] BEM++, `https://bempp.com/`.

[3] S. Kurepa, *Convex functions*, Glasnik Mat.-Fiz. Astr. Ser. II **11** (1956), br. 2, 89–93.

[4] ———, *Funkcionalna analiza*, Školska Knjiga, 1981.

Sažetak

CDSADAS dasd asd as

Summary

In this ...

Životopis

Dana ...