

2. domača naloga

Peter Rotar

Julij 2023

1. Naloga

Ideja: Da ne preobremenimo prostor se sprva naslonimo na pravila distributivnosti in asociativnosti algebraičnega prostora. Spodnji zapis prikazuje vrsti red množenja v *Matlabu*

$$Ax = \alpha Qx + \frac{\alpha}{n}e(d^T x) + \frac{1-\alpha}{n}e(e^T x).$$

Implementacija: Zgornjo idejo uporabimo pri implementaciji skript *Arnoldi.m* in *potencna_metoda.m*. Za optimalnost bi bilo potrebno Arnoldijevo metodo nadgraditi s ponovnim zagonom – v tem primeru to ni potrebno. Pri prvi poračunamo dominantno Ritzovo vrednost na matriki manjše velikosti od prvotne. V drugi poračunamo dominantno l. vr. s pomočjo Rayleighovega kvocienta. Skripte se izvajajo preko *nal1.m*.

Rezultati:

metoda	l. vr.	cas
eigs	1.000000	1.326550s
potencna	1.000000	0.626235s
arnoldi	1.000000	0.014667s

Table 1: Rezultati za $\alpha = 0.8$

Rezultati so prikazani v *Matlabu*. Sprva opazimo, da sprememba faktorja teleportacije α ne vpliva na dominantno l. vr. matrike A . Z večanjem α se hitrost konvergence pri **eigs** in potenčne metode poveča.

2. Naloga

Ideja: Sprva sestavimo 'lik', nato točke znotraj lika primerno oštevilčimo z $1, 2, 3, \dots$. Dobljeno matriko primerno poračunamo, da dobimo željene rezultate.

Implementacija: Kot ime narekuje, skripta *sestavi_boben.m* vrne boben v željeni oštevilčeni obliki. Skripta *bobni.m* uporabi ukaz **delsq** in popravi dobljeni matriki diagonalo. Skripta *lastne_funkcije.m* izriše enega izmed petih lastnih parov (λ, v) najmanjših lastnih vrednosti. Skripte se izvajajo preko *nal2.m*.

Rezultati: Rezultati so prikazani v *Matlabu*. Rešitev pokrije primere za $\lambda \in \{-342, 0, 574\}$ in $n \in \{12, 102, 1002\}$ – tj. 9 primerov z 18 prikazi lastnih funkcij za desni in levi boben. Opazimo, ko večamo n , se razlika med lastnimi vrednostmi levega in desnega bobna manjša.

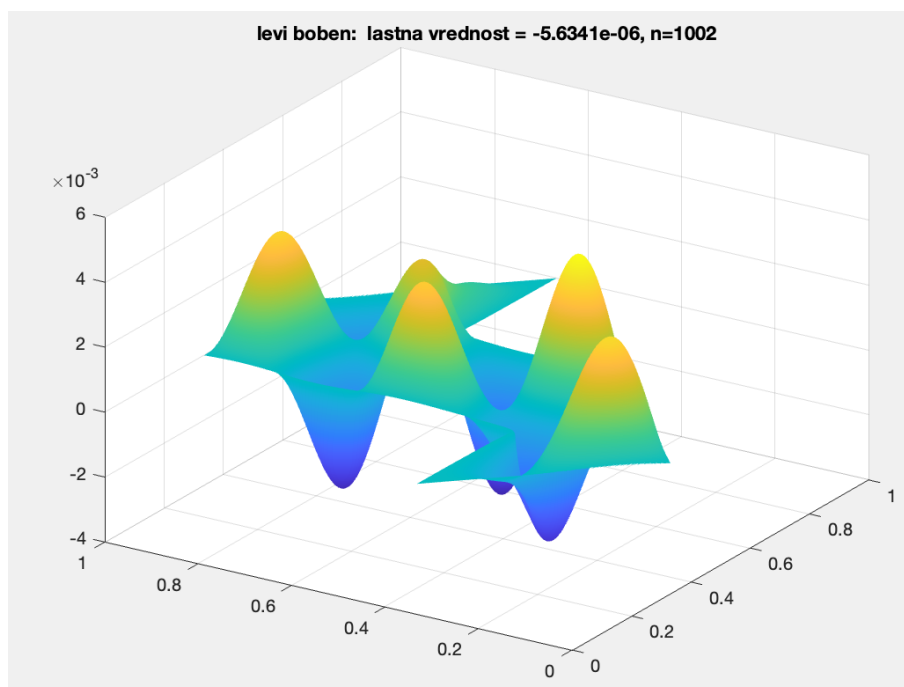


Figure 1: Lastna funkcija levega bobna za $n = 1002$.

```
lambda = 574.000000, n = 1002.000000
desni boben l. vr: -6.26316e-06 3.26054e-05 5.8488e-05 -6.0841e-05 0.000105248
levi boben l. vr: -5.63407e-06 3.30866e-05 5.79252e-05 -6.17955e-05 0.000105214

maksimalna razlika: 0.000001
```

Figure 2: Lastne vrednosti levega in desnega bobna za $\lambda = 574$ in $n = 1002$.

3. Naloga

Ideja: Zaradi predpostavk se lahko omejimo zgolj na presek poltraka s krajiščem v izhodišču z diksom. Dana je navadna diferencialna enačba $u''(r) + \frac{1}{r}u'(r) = f(r)$ za $r \in [0, R]$. Z uporabo diferenčne metode prostor diskretiziramo na $r_i = r_0 + ih = ih$, za $h = \frac{R}{n}$ in $i = 0, 1, \dots, n$. Tako dobimo:

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} + \frac{u_{i+1} - u_{i-1}}{2hr_i} = f_i.$$

Enačbo kanček preuredimo in dobimo:

$$2u_i - u_{i+1}\left(1 + \frac{h}{2r_i}\right) - u_{i-1}\left(1 - \frac{h}{2r_i}\right) = -h^2 f_i, \quad (1)$$

ki velja za $0 < i < n - 1$. Upoštevajoč $u_{-1} = u_1$ za $i = 0$ dobimo

$$2u_0 - 2u_1 = -h^2 f_0. \quad (2)$$

Zaradi robnega pogoja $u(R) = 0$, dobimo za $i = n - 1$ naslednji izraz

$$2u_{n-1} - u_{n-2}\left(1 - \frac{h}{2r_{n-1}}\right) = -h^2 f_{n-1}. \quad (3)$$

Upoštevajoč (1),(2) in (3) opazimo, da v našem sistemu $Ax = b$ matrika A predstavlja tridiagonalno matriko, kjer se vsaka vrstica, razen zadnje, sešteje v 0. Sledi, da je A š.d.d po vrsticah.

Implementacija: Zgornjo idejo uporabimo, da implementiramo tridiagonalno matriko. Pri implementaciji pazimo, da indekse pravilno zamaknemo. Pri iskanju latnih vrednosti se poslužimo prirejenim skriptam iz 1. naloge – uporabimo *Arnoldi.m*, *eigs_fun.m* in *potencna_metoda.m*. Hkrati sta dodani skripti *inv_potencna_metoda.m* za iskanje najmanjše lastne vrednosti in *sestavi_matriko.m*, kjer sestavimo željeno matriko za dani problem. Skripte se izvajajo preko *nal3.m*. Pozor: za $n = 10000$ je čas izvajanja daljši.

Rezultati: Opazimo, da je matrika A tridiagonalna. Hkrati opazimo, da je šibko diagonalno dominantna tj. $|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|$, razen za eno vrstico velja stroga neenakost (v našem primeru je to zadnja vrstica). Poračunane lastne vrednosti so prikazane v *Matlabu*. Iz rezultatov ugotovimo, da se l. vr. nahajajo znotraj intervala $[0, 4]$. Za $n = 10000$ funkcija **eigs** ne konvergira pri iskanju največjih lastnih vrednosti. To pomeni, da so si l. vr. zelo blizu ali pa so večkratne. Ker $\lambda_n \rightarrow 0$ in $\lambda_1 \rightarrow 4$ sledi da $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$ narašča, ko narašča n – torej je matrika A občutljiva.

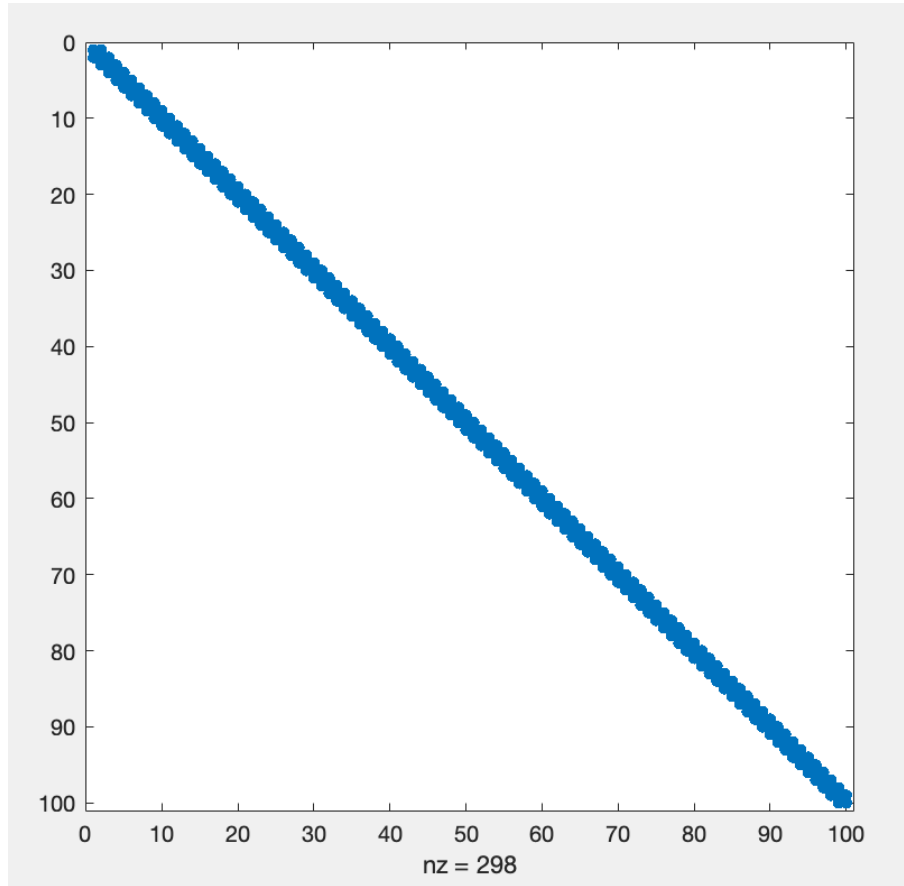


Figure 3: Prikaz nahajanja vrednosti v matriki A velikosti 100×100 .

```

N = 1000.000000
potencna = 3.999970 | inv. potencna = 0.000006
Arnoldi (max) = 3.99999 3.99997 3.99993 3.99986 3.99978 3.99967 3.99955 3.99941
| Arnoldi (min) = 5.78318e-06 3.04711e-05 7.48863e-05 0.000139038 0.000222927 0.000326552 0.000449913 0.000593008
Eigs (max) = 3.99999 3.99997 3.99993 3.99986 3.99978 3.99967 3.99955 3.99941
| Eigs (min) = 5.78318e-06 3.04711e-05 7.48863e-05 0.000139038 0.000222927 0.000326552 0.000449913 0.000593008

Pogojenostno število: 810609.978875

```

Figure 4: Rezultati metod za iskanje nekaj največjih in najmanjših l. vr. matrike A za $n = 1000$.

4. Naloga

Ideja: Naj bo $x = [u_{11}v_{11} \dots u_{nn}v_{nn}] \in \mathbb{R}^N$, kjer je $N = 2n^2$. Funkcijo f lahko smatramo kot preslikavo iz \mathbb{R}^N v \mathbb{R}^N , ki preslika vektor velikosti N v vektor velikosti N – torej, $f(x) = \dot{x} = [\dot{u}_{11}\dot{v}_{11} \dots \dot{u}_{nn}\dot{v}_{nn}]$. Spomnimo se Jacobijeve matrike, ki se za (i,j) -to mesto glasi $\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j}$. Ravno to idejo uporabimo pri generiranju iskane Jacobijeve matrike. Naj bo $f_{ij} := \dot{u}_{ij}$ in $g_{ij} := \dot{v}_{ij}$. Sledi, da $(u_{1,1}v_{1,1} \dots u_{n,n}v_{n,n})$ predstavljajo spremenljivke in $(f_{11}, g_{11} \dots f_{nn}, g_{nn})$ komponente funkcije f . Poračunamo Jacobijevo matriko v odvisnosti od f in na koncu upoštevamo $u_{ij} = C$ in $v_{ij} = B/C \ \forall i, j \in [n]$:

$$\begin{aligned}\frac{\partial f_{ij}}{\partial u_{i-1,j}} &= \frac{D_u}{(Lh)^2} \\ \frac{\partial f_{ij}}{\partial u_{i+1,j}} &= \frac{D_u}{(Lh)^2} \\ \frac{\partial f_{ij}}{\partial u_{i,j-1}} &= \frac{D_u}{(Lh)^2} \\ \frac{\partial f_{ij}}{\partial u_{i,j+1}} &= \frac{D_u}{(Lh)^2} \\ \frac{\partial f_{ij}}{\partial u_{ij}} &= \frac{-4D_u}{(Lh)^2} + B - 1 \\ \frac{\partial f_{ij}}{\partial v_{ij}} &= C^2.\end{aligned}$$

Podobno za g_{ij} dobimo:

$$\begin{aligned}\frac{\partial g_{ij}}{\partial v_{i-1,j}} &= \frac{D_v}{(Lh)^2} \\ \frac{\partial g_{ij}}{\partial v_{i+1,j}} &= \frac{D_v}{(Lh)^2} \\ \frac{\partial g_{ij}}{\partial v_{i,j-1}} &= \frac{D_v}{(Lh)^2} \\ \frac{\partial g_{ij}}{\partial v_{i,j+1}} &= \frac{D_v}{(Lh)^2} \\ \frac{\partial g_{ij}}{\partial v_{ij}} &= \frac{-4D_v}{(Lh)^2} - C^2 \\ \frac{\partial g_{ij}}{\partial u_{ij}} &= -B.\end{aligned}$$

Pri računanju upoštevamo, da velja $u_{0i} = u_{i0} = v_{0i} = v_{i0} = 0$ in $u_{n+1,i} = u_{i,n+1} = v_{n+1,i} = v_{i,n+1} = 0$ za vsak $i \in [n]$. Tako dobimo iskano razpršeno Jacobijevo matriko.

Implementacija: V imeniku **nal4** se nahajajo datoteke *sestavi_Jac_matriko_bf.m*, *sestavi_Jac_matriko.m*, *bisekcija.m* in *nal4.m*. Prva od naštetih je namenjena pri preverjanju pravilnosti druge za matrike manjše dimenzije. Druga po vrsti sestavi matriko s pomočjo ukaza **spdiags** in vrednosti, ki so poračunane zgoraj. Tretja s pomočjo bisekcije iterativno poišče iskani L . Skripte se izvajajo preko *nal4.m*.

Rezultati:

N	L
8	0.6924
18	0.7061
200	0.7236
242	0.7236
20000	0.7256

Table 2: Rezultati