METODE NUMERICE: Tema #2
Vectori si valori proprii.

Termen de predare: 19 APRILIE 2015

Ultima actualizare: 7 aprilie

Titulari curs: *Florin Pop, George Popescu*

Responsabili Tema: **Sorin N. Ciolofan, Clementin Cercel, Adelina Vidovici**

Obiectivele Temei

Obiectivele generale ale acestei teme sunt:

- calcularea numarului de valori proprii ale unei matrice tridiagonale simetrice care sunt mai mici decat o valoare data;
- calcularea unei limite superioare, respectiv inferioare intre care sunt situate toate valorile proprii ale unei matrice tridiagonale simetrice;
- separarea in intervale a valorilor proprii ale unei matrice tridiagonale simetrice folosind metoda sirului lui Sturm;
- calcularea valorilor si a vectorilor proprii pentru o matrice tridiagonala simetrica.

Descriere Generala

Problema calculului valorilor proprii ale unei matrice simetrice oarecare poate fi redusa la calculul valorilor proprii ale unei matrice tridiagonale simetrice care are aceleasi valori proprii (o astfel de matrice se poate obtine utilizand transformarea Householder).

O matrice tridiagonala simetrica $A \in \mathbb{R}^{n \times n}$, $A = \{a_{ij}\}$ contine elemente nenule pe diagonala principala si pe diagonalele adiacente (numite supradiagonala, respectiv subdiagonala) cu diagonala principala. Celelalte elemente dintr-o matrice tridiagonala simetrica sunt nule, adica $a_{ij} = 0$ pentru $|i - j| > 1$. Mai mult, o matrice tridiagonala simetrica are proprietatea ca $a_{ij} = a_{ji}$ pentru $|i - j| = 1$. Cu alte cuvinte, supradiagonala este egala cu subdiagonala.

Se constata ca o matrice tridiagonala simetrica este o matrice rara. Prin urmare, o matrice tridiagonala simetrica se pastreaza eficient in memorie folosind doi vectori: un vector reprezentand diagonala principala si celalalt vector reprezentand supradiagonala. Luand in calcul aceasta observatie, vom considera, in cele ce urmeaza, o matrice tridiagonala simetrica $A \in \mathbb{R}^{n \times n}$ de forma:

$$A = \begin{bmatrix} d_1 & s_1 & & & & \\ s_1 & d_2 & s_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & s_{n-2} & d_{n-1} & s_{n-1} \\ & & & & s_{n-1} & d_n \end{bmatrix} \tag{1}$$

Valorile proprii asociate cu matricea tridiagonală simetrica A pot fi obtinute calculand radacinile polinomului caracteristic $P_n(\lambda)$ definit prin relatia:

$$P_n(\lambda) = \left|A - \lambda I\right| = \begin{vmatrix} d_1-\lambda & s_1 & & & & \\ s_1 & d_2-\lambda & s_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & s_{n-2} & d_{n-1}-\lambda & s_{n-1} \\ & & & & s_{n-1} & d_n-\lambda \end{vmatrix} \tag{2}$$

Relatia 2 nu consitutie o metoda eficienta de a determina polinomul caracteristic $P_n(\lambda)$ deoarece calcularea determinatului implica efectuarea de multe operatii. Polinomul caracteristic $P_n(\lambda)$ poate fi calculat, in mod eficient, folosind urmatoarea relatie de recurenta:

$$P_i(\lambda) = (d_i - \lambda)P_{i-1}(\lambda) - s_{i-1}^2 P_{i-2}(\lambda), \quad i = 2:n \tag{3}$$

unde $P_0(\lambda) = 1$ si $P_1(\lambda) = d_1$ - λ.

Sirul de polinoame $P_0(\lambda), P_1(\lambda), P_2(\lambda), ..., P_n(\lambda)$ formeaza un sir Sturm ([1]). Enuntam, in continuare, o proprietatea asociata unui sir Sturm care ne permite sa determinam numarul de valori proprii asociate cu matricea tridiagonala simetrica A, mai mici decat un numar real dat.

Proprietate sir Sturm: Fie sirul Sturm $P_0(\lambda)$, $P_1(\lambda)$, $P_2(\lambda)$, ..., $P_n(\lambda)$ si o valoare α. Consideram valoarea polinoamelor din acest sir Sturm pentru α ca fiind $P_0(\alpha)$, $P_1(\alpha)$, $P_2(\alpha)$, ..., $P_n(\alpha)$. Numarul radacinilor ecuatiei $P_n(\lambda) = 0$, mai mici decat α, este egal cu numarul schimbarilor de semn ale sirului numeric $P_0(\alpha)$, $P_1(\alpha)$, $P_2(\alpha)$, ..., $P_n(\alpha)$.

Observatie 1: In stabilirea numarului de schimbari de semn ale sirului numeric $P_0(\alpha)$, $P_1(\alpha)$, $P_2(\alpha)$, ..., $P_n(\alpha)$ se compara $P_i(\alpha)$ cu $P_{i+1}(\alpha)$ pentru i = 0 : n-1.

Observatie 2: Daca $P_i(\alpha) = 0$, semnul pentru $P_i(\alpha)$ se va considera ca fiiind semnul opus lui $P_{i-1}(\alpha)$.

Urmatoarea teorema, datorata lui Gershgorin ([2]), ne permite sa obtinem o limita inferioara λ_{min} si una superioara λ_{max} intre care sunt situate toate valorile proprii ale unei matrice tridiagonale simetrice. Cu alte cuvinte, toate valorile proprii sunt cuprinse in intervalul $[\lambda_{min},\ \lambda_{max}]$. Vom enunta aceasta teorema pentru cazul general cand matricea patratica este de forma oarecare, dar teorema ramane valabila si pentru o matrice tridiagonala simetrica.

Teorema lui Gershgorin: Fie $B \in \mathbb{R}^{n\times n}$, $B = \{b_{ij}\}$ o matrice simetrica de forma oarecare. Limita inferioara λ_{min} si cea superioara λ_{max} a valorilor proprii ale matricei B sunt date de relatiile:

$$\lambda_{min} = \min_{i=1:n}(b_{ii} - t_i), \qquad \lambda_{max} = \max_{i=1:n}(b_{ii} + t_i) \tag{4}$$

unde

$$t_i = \sum_{j=1, j\neq i}^{n} |b_{ij}| \tag{5}$$

Cerinta 1 (15p)

Scrieti o functie numita *ValoriPolinoame* care calculeaza valoarea fiecarui polinom din sirul de polinoame $P_0(\lambda)$, $P_1(\lambda)$, $P_2(\lambda)$, ..., $P_n(\lambda)$ pentru o valoare reala λ data. Implementarea acestei functii se va face in fisierul *ValoriPolinoame.m*. Functia *ValoriPolinoame* are ca intrare urmatoarele argumente in aceasta ordine: diagonala principala $d = [d_1, d_2, ..., d_n]$, supradiagonala $s = [s_1, s_2, ..., s_{n-1}]$ si un numar real *val_lambda* pentru care se calculeaza valoarea polinoamelor $P_0(\lambda)$, $P_1(\lambda)$, $P_2(\lambda)$, ..., $P_n(\lambda)$. Functia are ca rezultat vectorul $P = [P_0(val\_lambda), P_1(val\_lambda), P_2(val\_lambda), ..., P_n(val\_lambda)]$ de lungime *n + 1*.

```
function P = ValoriPolinoame(d, s, val_lambda)
   % Intrari:
   % -> d:  diagonala principala a matricei tridiagonale simetrice;
   % -> s: supradiagonala matricei tridiagonale simetrice;
   % -> val_lambda: valoare pentru lambda.
   % Iesiri:
   % -> P: vectorul P = [P0(val_lambda) P1(val_lambda), ..., Pn(val_lambda)].
```

Listing 1: ValoriPolinoame.m

Cerinta 2 (15p)

Folosind functia *ValoriPolinoame* si proprietatea unui sir Sturm enuntata in sectiunea Descriere Generala, scrieţi o funcţie numita *NrValProprii* pentru a determina numarul de valori proprii ale unei matrice tridiagonale simetrice care sunt mai mici decat o valoare reala λ data. Implementarea functiei *NrValProprii* se va face in fisierul *NrValProprii.m*. Functia *NrValProprii* are ca intrare urmatoarele argumente in aceasta ordine: *d*, *s* si *val_lambda* (cu aceeasi interpretare ca la Cerinta 1). Functia are ca rezultat un numar natural reprezentand numarul de valori proprii ale matricei tridiagonale simetrice mai mici decat *val_lambda*.

```
function numvp = NrValProprii(d, s, val_lambda)
   % Intrari:
   % -> d: diagonala principala a matricei tridiagonale simetrice;
   % -> s: supradiagonala matricei tridiagonale simetrice;
   % -> val_lambda: valoare pentru lambda.
   % Iesiri:
   % -> numvp: numarul de valori proprii mai mici decit val_lambda.
```

Listing 2: NrValProprii.m

Cerinta 3 (15p)

Folosind teorema lui Gershgorin enuntata in sectiunea Descriere Generala, scrieţi urmatoarea funcţie numita *LimiteValProprii* pentru a determina limitele incadratoare (limita inferioara, respectiv superioara) a valorilor proprii ale unei matrice tridiagonale simetrice. Implementarea functiei *LimiteValProprii* se va face in fisierul *LimiteValProprii.m*. Functia *LimiteValProprii* are ca intrare urmatoarele argumente in aceasta ordine: *d* si *s* (cu aceeasi interpretare ca la Cerinta 1). Functia are ca rezultat limita inferioara si cea superioara intre care sunt cuprinse toate valorile proprii ale matricei tridiagonale simetrice.

```
function [limita_inf, limita_sup] = LimiteValProprii(d, s)
   % Intrari:
   % -> d: diagonala principala a matricei tridiagonale simetrice;
   % -> s: supradiagonala matricei tridiagonale simetrice.
   % Iesiri:
   % -> [limita_inf limita_sup]: limitele incadratoare pentru valorile proprii ale
     matricei tridiagonale simetrice.
```

Listing 3: LimiteValProprii.m

Cerinta 4 (15p)

Folosind functiile $LimiteValProprii$ si $NrValProprii$, scrieti o funcţie numita *IntervaleValProprii* pentru a separa cele mai mici m valori proprii asociate cu o matrice tridiagonala simetrica. Separarea celor mai mici m valori proprii presupune determinarea unui sir numeric r_1, r_2, ..., r_m, r_{m+1} astfel incat:

- $r_1 < r_2 < ... < r_m < r_{m+1}$;
- Fiecare interval $(r_i,\ r_{i+1})$, $i = 1 : m$ contine o singura valoare proprie λ_i astfel incat cea mai mica valoarea proprie a matricei tridiagonale simetrice se afla in intervalul $(r_1,\ r_2)$, urmatoarea valoare proprie cea mai mica se afla in intervalul $(r_2,\ r_3)$ s.am.d.

Implementarea functiei *IntervaleValProprii* se va face in fisierul *IntervaleValProprii.m*. Functia *IntervaleValProprii* are ca intrare urmatoarele argumente in aceasta ordine: *d*, *s* (cu aceeasi interpretare ca la Cerinta 1) si *m*.

```
function r = IntervaleValProprii(d, s, m)
    % Intrari:
    % -> d: diagonala principala a matricei tridiagonale simetrice;
    % -> s: supradiagonala matricei tridiagonale simetrice;
    % -> m: limita pentru numarul de valori proprii.
    % Iesiri:
    % -> r: un vector r = [r1, r2, ..., rm, rm+1] de dimensiune m+1.
```

Listing 4: IntervaleValProprii.m

Indicatie: Elementele sirului r se calculeaza in aceasta ordine: r_1, apoi r_{m+1}, r_m, r_{m-1} s.a.m.d. pana se calculeaza si r_2. Mai intai, folosind functia $LimiteValProprii$, se determina limita inferioara *limita_inf* si cea superioara *limita_sup* intre care se afla toate valorile proprii (inclusiv cele mai mici m valori proprii) ale matricei tridiagonale simetrice. Elementul r_1 se va initializa cu valoarea *limita_inf*. Mai mult, se introduce o variabila auxiliara r_{m+2} cu valoarea *limita_sup*. Variabila auxiliara r_{m+2} este folosita doar pentru a calcula elementul r_{m+1} si nu trebuie salvata in vectorul r.

Pentru determinarea elementelor r_{k+1}, $k = m : -1 : 1$ se parcurg urmatoriii pasi:

1. Se calculeaza mijlocul intervalului $[r_1,\ r_{k+2}]$ folosind relatia:
$$mij = (r_{k+2} + r_1)/2 \tag{6}$$
2. Se calculeaza lungimea intervalului $[mij,\ r_{k+2})$ folosind relatia :
$$h = r_{k+2} - mij = r_{k+2} - (r_1 + r_{k+2})/2 = (r_{k+2} - r_1)/2 \tag{7}$$
3. Se calculeaza numarul de valori proprii mai mici decat valoarea *mij*
$$numvp = NrValProprii(d, s, mij) \tag{8}$$
4. Se actualizeaza valoarea *mij* astfel:
   4.1. $h = h/2$
   4.2. daca $numvp < k$ atunci
   4.3. $\quad mij = mij + h$
   4.4. altfel daca $numvp > k$ atunci
   4.5. $\quad mij = mij - h$
5. Se repeta algoritmul de la pasul 3 pana cand $numvp = k$.
6. Se obtine astfel valoarea petru r_{k+1}:
$$r_{k+1} = mij \tag{9}$$

Cerinta 5 (15p)

Folosind functiile *IntervaleValProprii*, *NrValProprii*, *ValoriPolinoame* si metoda bisectiei (studiata in Laboratorul 6), implementati o functie numita *CalculezValProprii* pentru a determina cu o anumita precizie cele mai mici m valori proprii ale unei matrice tridiagonale simetrice. Implementarea functiei *CalculezValProprii* se va face in fisierul *CalculezValProprii.m*. Functia *CalculezValProprii* are ca intrare urmatoarele argumente in aceasta ordine: d, s, m (cu aceeasi interpretare ca la Cerinta 4) si precizia *eps* folosita in metoda bisectiei. Rezultatul functiei este un vector $vp = [\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m]$ de dimensiune m, care contine cele mai mici m valori proprii.

```
function vp = CalculezValProprii(d, s, m, eps)
    % Intrari:
    % -> d: diagonala principala a matricei tridiagonale simetrice;
    % -> s: supradiagonala matricei tridiagonale simetrice;
    % -> m: limita pentru numarul de valori proprii;
    % -> eps: precizia determinarii.
    % Iesiri:
    % -> vp: vector de valori proprii, de dimensiune m.
```

Listing 5: CalculezValProprii.m

Cerinta 6 (15p)

Folosind metoda puterii inverse, scrieţi o funcţie numita *PutereInv* pentru a determina valoarea proprie cea mai apropiata de o deplasare data, precum si vectorul propriu corespunzator acestei valori proprii. Implementarea functiei *PutereInv* se va face in fisierul *PutereInv.m*. Functia *PutereInv* are ca intrare urmatoarele argumente in aceasta ordine: d, s (cu aceeasi interpretare ca la Cerinta 1), valoarea de deplasare h, aproximatia initiala *y0* a vectorului propriu, numarul maxim de iteratii *maxIter* si precizia *eps*. Functia *PutereInv* are ca rezultat o valoarea proprie si un vector propriu cu semnificatia descrisa in cerinta. Rezultatele vor fi generate cind fie se va atinge precizia *eps* sau numarul maxim de iteratii *maxIter*.

Observatii: In metoda puterii inverse pe care o aveti de implementat, trebuie sa lucrati cu cei doi vectori d si s pe care functia *PutereInv* ii primeste ca intrare si nu trebuie sa compuneti matricea A din vectorii d si s. Cerinta 6 nu se va puncta, indiferent de rezultatele functiei *PutereInv*, daca in metoda puterii inverse compuneti matricea A si veti lucra cu aceasta matrice. Pentru a calcula solutia unui sistem de ecuatii liniare, unde matricea sistemului este tridiagonala si simetrica, se va folosi algoritmul Thomas din Laboratorul 4.

```
function [valp vecp] = PutereInv(d, s, h, y, maxIter, eps)
    % Intrari:
    % -> d: diagonala principala a matricei tridiagonale simetrice;
    % -> s: supradiagonala matricei tridiagonale simetrice;
    % -> y: aproximatia initiala a vectorului propriu;
    % -> h: deplasare;
    % -> maxIter: numarul maxim de iteratii;
    % -> eps: precizia determinarii.
    % Iesiri:
    % -> [valp vecp]: valoarea proprie valp, respectiv vectorul propriu vecp.
```

Listing 6: PutereInv.m

Detalii de implementare si redactare

Pentru implementarea temei, functiile mentionate in enuntul temei sunt obligatorii si trebuie implementate conform antetelor prezentate mai sus, fiecare functie in fisierul sau (vor exista deci minimum sase fisiere

.m). Nu este permisa folosirea in implementarea cerintelor 1-6 a unor functii predefinite din Octave pentru calculul valorilor proprii/vectorilor proprii (cum este functia *eig()*). De asemenea, trebuie sa tineti cont de urmatoarele aspecte:

- Codul sursa va contine comentarii semnificative si sugestive cu privire la implementarea cerintelor.
- Existenta unui fisier `README` care va prezenta detaliile legate de implementarea temei.
- Fisierele sursa **.m* folosite in rezolvarea temei impreuna cu fisierul `README` vor fi incluse intr-o arhiva `.zip`. Denumirea arhivei respecta specificatiile din regulamentul cursului.
- Tema se va implementa in Octave.
- Se acorda 90 de puncte pentru o tema care functioneaza conform cerintelor descrise mai sus. 10 puncte se acorda pentru fisierul `README` scris corespunzator.

Resurse Web

1. http://en.wikipedia.org/wiki/Sturm's_theorem
2. http://www.encyclopediaofmath.org/index.php/Gershgorin_theorem