Laboratorul 7 – Vectori și valori proprii

Responsabil: Tălău Cristian<cristi.talau@gmail.com>

Data publicării: 21.05.2009

Data ultimei modificări: 21.05.2009

Obiective

În urma parcurgerii acestui laborator studentul va fi capabil să:

- folosească funcțiile disponibile în OCTAVE pentru calcularea vectorilor și valorilor proprii ale unei matrici.
- implementeze algoritmi iterativi de calcul al vectorilor și valorilor proprii.
- reprezinte grafic evoluția unor algoritmi de calcul al vectorilor și valorilor proprii.

Aplicaţii

1. Folosirea functiilor OCTAVE pentru caculul vectorilor și valorilor proprii

Considerăm problema aflării rădacinilor unui polinom cu *coeficienți* reali. Se știe ca aceasta se reduce la aflarea valorilor proprii ale *matricei companion polinomial*.

Hint: Pentru construirea matricei companion puteți folosi urmatoarea secventă de cod.

```
p = p/p(1);  % normăm polinomul
n = length(p);
comp = diag(ones(1,n-2),-1);   % punem 1 sub diagonala
comp(:,n-1) = -p(n:-1:2)'; % pe ultima coloană coeficienții cu semn schimbat
```

Mediul de programare OCTAVE furnizează o funcție de calcul al vectorilor și valorilor proprii pentru o matrice A arbitrară. Un exemplu de utilizare:

```
[V, lambda] = eig(A);
```

unde:

- *A* este matricea căreia vrem să îi calculăm perechile proprii. *A* trebuie să fie matrice patrată, altfel nu au sens noțiunile de vector și valoare proprie.

- *V* este o matrice ale cărei coloane sunt vectorii proprii ai matricei A.

- *lambda* este o matrice care are pe diagonală valorile proprii ale matricei A, și in rest 0.

Creați o funcție care să calculeze rădăcinile unui polinom și care să aibă semnătura:

```
Function r = myroots( poly)
% poly – coeficienții polinomului căruia vrem să îi calculăm rădăcinile
%        primul coeficient este cel al puterii celei mai mari a lui X
```

Pentru validarea rezultatelor folosiți funcția predefinită *roots (help roots)*.

2. Reprezentarea grafică a evolutiei metodei puterii directe - 2p

Scrieți o funcție OCTAVE care să aplice metoda puterii directe matricei *A* de dimensiune 2, iterațiile vor porni simultan cu *n* vectori. Aceștia sunt coloanele unei matrice *v* creeată astfel:

```
n=100; t = linspace(0,2*pi,n);
v = [cos(t); sin(t)];
```

adică vectorii de pozitie ai varfurilor unui poligon regulat înscris în cercul unitate.

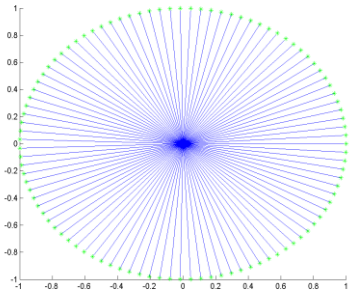
Programul va trebui să deseneze setul curent de vectori, după fiecare reactualizare a acestuia (calcul conform relației de recurență și normare la unitate) si să astepte apăsarea unei taste pentru continuare.

Pentru desenarea unui set de vectori se va folosi următoarea secvență de cod:

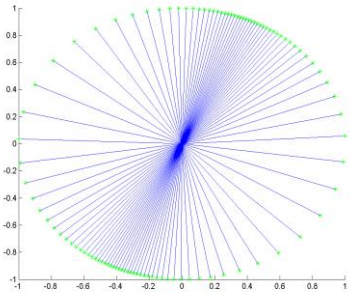
```
clf;hold on;
for j = 1:length(v)
 plot([0, v(1,j)],[0, v(2,j)] ,'b-', v(1,j), v(2,j),'g*');
end
```

Desenați vectorul propriu asociat valorii proprii dominante. Puteți folosi următoarea secvență de cod:

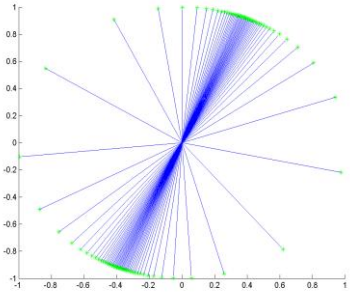
```
[v,l]=eig(A);
lmax = max(diag(l));                  % v.p. dominanta
vec = v(:,find(~(diag(l)-lmax)));     % vectorul asociat
plot( [0 vec(1)],[0 vec(2)], 'r-', vec(1), vec(2),'r*');
```



1.Initial



2.Dupa o iteratie



3.Dupa patru iteratii

Se observă că dupa câteva iterații, (aproape) toți vectorii vor avea aceeași direcție: direcția vectorului propriu căutat. Exceptie fac eventualii vectori care inițial sunt perpendiculari pe vectorul propriu atașat valorii proprii dominante.

Hint: Pentru a obține vectorii de la un pas, se înmulțește matricea A cu vectorii de la pasul precedent și se normează vectorii obținuți.

3. Aflarea valorilor și vectorilor proprii intermediari – metoda defalției

Cu ajutorul metodei puterii se poate găsi valoarea proprie dominantă si vectorul propriu corespunzător.

Pentru a afla restul de perechi proprii vom folosi urmatoarea constucție:

- Inițial avem o matrice A cu vectorii proprii $\{x_1, x_2, \dots, x_n\}$ și valorile proprii $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ dintre care pe λ_1 și x_1 îi calculăm cu metoda puterii.
- Contruim matricea $B = A - \lambda_1 x_1 y^T$, unde $y = \frac{A(1,:)}{x_1(1)\lambda_1}$. Această matrice va avea valorile proprii $\{0, \lambda_2, \lambda_3, \dots, \lambda_n\}$ și vectorii proprii $\{x_1, x_1 - x_2, x_1 - x_3, \dots, x_1 - x_n\}$.
- Dacă din matricea *B* scoatem prima linie și prima coloană, își păstrează perechile proprii mai puțin prima. Pe această matrice putem aplica același procedeu.

Scrieți o funcție OCTAVE care să calculeze toate valorile și vectorii proprii pentru o matrice folosind construcția de mai sus. Funcția trebuie să aibă următoarea semnătură:

```
function [v,lambda] = myeig(A)
```

Hint: Observați ce valori are prima componentă a vectorilor proprii ai matricei B.

Pentru testare trebuie să vă asigurați ca matricele au valori proprii reale. Puteți sa folosiți matrice simetrice și pozitiv-definite. Pentru a genera random astfel de matrice puteți folosi următoarea secvență:

```
n = 4;
A = floor( 10 * rand(n));
A = A*A'+eye(n);
```

Bonus: Observați în construcția matricei B, o împarțire la $x_1(1)$. Propuneți un algoritm care sa funcționeze și când aceast element este 0. Hint: Gândiți-va la soluția folosită pentru metoda Gauss într-o situație similară.