Ιδιοτιμές και ιδιοδιανύσματα

Έστω A ένας τετραγωνικός πίνακας n x n και u διάνυσμα στήλη με n στοιχεία και ας θεωρούμε την εξίσωση

$$A\,u = \lambda\,u$$

Ο αριθμός λ λέγεται ιδιοτιμή του A και το αντίστοιχο διάνυσμα u ιδιοδιάνυσμα του A (που αντιστοιχεί στην ιδιοτιμή λ).

- Προφανώς το u ανήκει στον πυρήνα του (A-λ I).
- Στο Matlab οι ιδιοτιμές και τα αντίστοιχα ιδιοδιανύσματα ενός πίνακα A βρίσκονται με την εντολή **[v,d] = eig(A)**

  Το v είναι ένας πίνακας n x n του οποίου οι στήλες είναι τα ιδιοδιανύσματα και το d είναι ένας διαγώνιος πίνακας που τα στοιχεία της διαγωνίου περιέχουν τις αντίστοιχες ιδιοτιμές.
- Για παράδειγμα:

```
>> A=[1 5;0 -2];

>> [v,d] = eig(A)

v =

   1.0000   -0.8575

        0    0.5145

d =

     1     0

     0    -2
```

```
>> A*v(:,1)-d(1,1)*eye(2)*v(:,1)

ans =

     0

     0

>> A*v(:,2)-d(2,2)*eye(2)*v(:,2)

ans =

    0

    0
```

- **Θεώρημα:** Έστω ότι $\lambda_1,\ldots,\lambda_m$ διαφορετικές ιδιοτιμές με αντίστοιχα ιδιοδιανύσματα $u_1,\ldots,u_m$. Τότε τα $u_1,\ldots,u_m$ είναι γραμμικώς ανεξάρτητα.

```
>> linind(v(:,1)',v(:,2)')

 Vectors are linearly independend.
```

Κάθε n x n πίνακας A έχει το πολύ n το πλήθος ιδιοτιμές.

Η πολλαπλότητα της ιδιοτιμής λ ορίζεται ώς η διάσταση του υπόχωρου των γενικευμένων ιδιοδιανυσμάτων που αντιστοιχούν στην ιδιοτιμή λ, με άλλα λόγια

- Πολλαπλότητα της λ = dim null(A - λ I)^n

Παράδειγμα,

```
>> A=[6 7 7;0 6 7;0 0 7];

>> null((A-6*eye(3))^3)

ans =

     0    -1

    -1     0

     0     0

>> null((A-7*eye(3))^3)

ans =

   -0.9921

   -0.1240

   -0.0177
```

Το χαρακτηριστικό και ελάχιστο πολυώνυμο

Έστω A μιγαδικός τετραγωνικός πίνακας n x n και έστω $\lambda_1,\ldots,\lambda_m$ διαφορετικές ιδιοτιμές του A. Έστω d_k η πολλαπλότητα της ιδιοτιμής λ_k του A. Το πολυώνυμο

$$(z-\lambda_1)^{d_1} \cdots (z-\lambda_m)^{d_m}$$

λέγεται το χαρακτηριστικό πολυώνυμο του A. Οι ρίζες του χαρακτηριστικού πολυωνύμου είναι οι ιδιοτιμές του A.

Στο Matlab το χαρακτηριστικό πολυώνυμο ενός τετραγωνικού πίνακα A βρίσκεται με την εντολή **poly**

```
>> xar=poly(A)
xar =
    1   -19   120  -252
>> u=[z^3 z^2 z 1];
>> polywnymo=dot(xar,u)
polywnymo =
z^3 - 19*z^2 + 120*z - 252
>> solve(polywnymo,z)
ans =
 6
 6
 7
```

Το θεώρημα Cayley-Hamilton

Θεώρημα Cayley-Hamilton: Έστω ένας nxn πίνακας A και q(z) το χαρακτηριστικό πολώνυμο του A. Τότε p(A)=0.

Η εντολή **polyvalm** υπολογίζει την τιμή ενός πολυωνύμου με την έννοια πινάκων!!

```
>> p=poly(A)

p =

    1   -19   120  -252

>> polyvalm(p,A)

ans =

    0     0     0

    0     0     0

    0     0     0
```

Το ελάχιστο πολυώνυμο

Υπάρχει ένα (μονικό) πολυώνυμο με ελάχιστο βαθμό τέτοιο ώστε όταν εφαρμόζεται στο πίνακα A δίνει τον μηδενικό πίνακα. Το πολυώνυμο αυτό λέγεται το ελάχιστο πολυώνυμο του A. Πράγματι τα

$$(I , A ,A^2 ,..., A^{(n^2)})$$

δεν μπορεί να είναι γραμμικώς ανεξάρτητα γιατί ο χώρος έχει διάσταση n^2 ενώ έχουμε n^2+1 πίνακες. Έστω m ο μικρότερος ακέραιος τέτοιος ώστε τα

$$(I , A ,A^2 ,..., A^m)$$

είναι γραμμικώς εξαρτημένα. Οπότε κάποιος πίνακας από την λίστα γράφεται ως γραμμικός συνδυασμός των υπολοίπων πινάκων στην λίστα. Επειδή m είναι ο μικρότερος τέτοιος ακέραιος συνάγουμε ότι ο πίνακας A^m γράφεται ως γραμμικός συνδυασμός των

$$(I , A , .. A^{(m-1)})$$

Οπότε υπάρχουν a_0, a_1,...,a_{m-1} πραγματικοί (μιγαδικοί) αριθμοί τέτοιοι ώστε

$$a_0 I + a_1 A + \dots + a_{m-1} A^{(m-1)} + A^m = 0$$

Το πολυώνυμο

$$a_0 + a_1 z + \ldots + a_{m-1} z^{(m-1)} + z^m = 0$$

λέγεται ελάχιστο πολυώνυμο του A. Είναι το ελάχιστο πολυώνυμο p για το οποίο ισχύει p(A)=0.

- Παράδειγμα το ελάχιστο πολυώνυμο του μοναδιαίου πίνακα I είναι το z-1

  Ο βαθμός του ελάχιστου πολυωνύμου ενός πίνακα nxn είναι το πολύ n

- **Θεώρημα:** Το ελάχιστο πολυώνυμο του A διαιρεί το χαρακτηριστικό πολυώνυμο του A.

- **Θεώρημα:** Οι ρίζες του ελάχιστου πολυωνύμου του A είναι ακριβώς οι ιδιοτιμές του A.

  Αλλά πως υπολογίζουμε το ελάχιστο πολυώνυμο ενός τετραγωνικού πίνακα A?

  Το Matlab **ΔΕΝ** έχει μια τέτοια εντολή έτοιμη.

- Δεν πειράζει, θα την φτιάξουμε εμείς!!!!!

```
function [minimal V] = minpoly(A)
% Βρίσκει το ελάχιστο πολυώνυμο ενός πίνακα A
% minimal είναι στην μορφή πολυωνύμου,
% V το ελάχιστο πολυώνυμο σε μορφή διανύσματος
[m,n] = size(A);
qu=[];
i=1;
mnm=reshape(eye(n).',1,[]);
[mqu,nqu] = size(qu);
while nqu==0
    mnm=[mnm;reshape((A^i).',1,[])];
    qu=null(sym(mnm)');
    i=i+1;
    [mqu,nqu] = size(qu);
end
i=i-1;
V=double(fliplr(qu.'));
vec=[1];
syms z
for k=1:i
   vec=[z^k vec];
end
minimal=dot(V,vec);
```

```
>> A=eye(4);A(1,2)=1;A(3,4)=1;
>> A
A =
     1     1     0     0
     0     1     0     0
     0     0     1     1
     0     0     0     1
>> [m,v]=minpoly(A)
m =
z^2 - 2*z + 1
v =
     1    -2     1
>> B=A;B(3,4)=0;
>> [m,v]=minpoly(B)
m =
z^2 - 2*z + 1
v =
     1    -2     1
```

```
>> C=B;C(4,4)=-1;
>> [m,v]=minpoly(C)
m =
z^3 - z^2 - z + 1
v =
     1    -1    -1     1

>> D=C;C(3,4)=-1;
>> [m,v]=minpoly(D)
m =
z^3 - z^2 - z + 1
v =
     1    -1    -1     1
```

- Οι πίνακες C και D έχουν το ίδιο ελάχιστο πολυώνυμο

```
>> poly(A)

ans =

    1    -4     6    -4     1

>> poly(B)

ans =

    1    -4     6    -4     1

>> poly(C)

ans =

    1    -2     0     2    -1

>> poly(D)

ans =

    1    -2     0     2    -1
```

- Οι πίνακες A και B έχουν το ίδιο χαρακτηριστικό πολυώνυμο καθώς επίσης και οι πίνακες C και D.

Το ελάχιστο πολυώνυμο διαιρεί το χαρακτηριστικό πολυώνυμο

Θεώρημα: Έστω τετραγωνικός πίνακας A διάστασης n x n και έστω πολυώνυμο p(x) τέτοιο ώστε P(A)=0. Τότε το ελάχιστο πολυώνυμο m(x) διαιρεί το p(x).

- Συνεπώς, αφού για το χαρακτηριστικό πολυώνυμο p(x) του A ισχύει p(A)=0, τότε το ελάχιστο πολυώνυμο m(x) διαιρεί το p(x).

Παράδειγμα, ας θεωρήσουμε τον προηγούμενου πίνακα A.

```
>> [pol,m]=minpoly(A)

pol =

z^2 - 2*z + 1

m =

     1    -2     1

>> p=poly(A)

p =

     1    -4     6    -4     1
```

- Η διαίρεση πολυωνύμων στο Matlab εκτελείται με την εντολή **deconv**

```
>> [q,r]=deconv(p,m)

q =

     1    -2     1

r =

     0     0     0     0     0
```

```
>> C
C =
     1     1     0     0
     0     1     0     0
     0     0     1    -1
     0     0     0    -1
>> [pol,m]=minpoly(C)
pol =
z^3 - z^2 - z + 1
m =
     1    -1    -1     1
>> p=poly(C)
p =
     1    -2     0     2    -1
>> [q,r]=deconv(p,m)
q =
     1    -1
r =
     0     0     0     0     0
```