

# Transformări ortogonale: Householder și Givens. Algoritmul Gram-Schmidt. Polinoame ortogonale

## Noțiuni teoretice

### Vectori ortogonali. Matrice ortogonală

Fie vectorii coloană  $x, y \in R^n$  de forma  $x = [x_1 \ x_2 \ \dots \ x_n]^T$  și  $y = [y_1 \ y_2 \ \dots \ y_n]^T$ . Definim:

- *produsul scalar*:  $\langle x, y \rangle = \sum_{i=1}^n x_i y_i = y^T x$ ;
- *norma euclidiană*:  $\|x\|_2 = \sqrt{\langle x, x \rangle} = \sqrt{x^T x} = \left( \sum_{i=1}^n x_i^2 \right)^{1/2}$ .

Spunem că:

- vectorii  $x$  și  $y$  sunt *ortogonali* dacă  $\langle x, y \rangle = 0$ ;
- vectorii  $x$  și  $y$  sunt *ortonormați* dacă sunt ortogonali,  $\|x\|_2 = 1$  și  $\|y\|_2 = 1$ .

O matrice  $H \in R^{n \times n}$  este *ortogonală* dacă are coloanele vectori ortonormați. Mai mult, o matrice ortogonală are proprietățile următoare:

1.  $H^T H = H H^T = I_n$ ;
2.  $H^{-1} = H^T$ ;
3.  $\|Hx\|_2 = \|x\|_2$ ;
4.  $\|H\|_2 = 1$ ;

5.  $\|HA\|_2 = \|A\|_2$ ;

6.  $\det(H) = \pm 1$ .

## Transformarea Householder

Metoda propusă inițial de Alston Scott Householder este folosită pentru a transforma o matrice  $A \in R^{n \times n}$  simetrică într-o matrice tridiagonală cu aceleași valori proprii. Prezentarea generală a metodei poate fi găsită la adresa <sup>1</sup>.

Pentru o matrice  $A \in R^{m \times n}$ , definim transformarea Householder  $R = HA$  folosind reflectori elementari Householder  $H_p$ , astfel încât:

$$H = H_{\min(m-1,n)} \dots H_p \dots H_2 H_1, \quad A = H^T R.$$

Un reflector elementar Householder  $H_p$  este dat de relația:

$$H_p = I_m - 2 \frac{v_p v_p^T}{v_p^T v_p}$$

unde:

- $v_p = [0 \ 0 \ \dots \ v_{pp} \ \dots \ v_{mp}]^T$  se numește vector Householder;
- $v_{pp} = a_{pp} + \sigma_p$ ;
- $v_{ip} = a_{ip}, \forall i > p$ ;
- $a_p = [a_{1p} \ a_{2p} \ \dots \ a_{pp} \ \dots \ a_{mp}]^T$  este coloana  $p$  din matricea  $A$ ;
- $\sigma_p = \text{sign}(a_{pp}) \sqrt{\sum_{i=p}^m a_{ip}^2}$ ;
- $\beta_p = \sigma_p v_{pp}$ .

Folosind un reflector elementar Householder putem aduce o matrice la forma superior triunghiulară astfel:

Datorită formei reflectorilor elementari Householder, înmulțirea  $A = H_p A$  se efectuează astfel:

- Coloanele  $1 : p - 1$  din matricea  $A$  rămân neschimbate;
- Coloana  $p$  din matricea  $A$  se modifică astfel:
  - elementele de pe liniile  $1 : p - 1$  rămân neschimbate;
  - $a_{pp} = -\sigma_p$ ;

---

<sup>1</sup><http://mathfaculty.fullerton.edu/mathews/n2003/HouseholderMod.html>

---

**Algorithm 1** Transformarea unei matrice la forma superior triunghiulară

---

```

1: procedure [Q, R] = ST(A)
2:   [m,n] = size(A);
3:   H = In;
4:   for p = 1 : min(m - 1, n) do
5:     Hp = In - 2  $\frac{v_p v_p^T}{v_p^T v_p}$ ;
6:     A = Hp A;
7:     H = Hp H;
8:   end for
9:   Q = HT;
10:  R = A;
11: end procedure

```

---

- $a_{ip} = 0, \forall i > p$ ;
- Coloanele  $j = p + 1 : n$  din matricea  $A$  se modifică astfel:
  - elementele de pe liniile  $1 : p - 1$  rămân neschimbate;
  - $a_{ij} = a_{ij} - \tau_j \cdot v_{ip}, \forall i \geq p, \tau_j = \frac{\sum_{i=p}^m v_{ip} \cdot a_{ij}}{\beta_p}$ .

## Transformarea Givens

Metoda Givens este folosită pentru a descompune o matrice  $A \in R^{m \times n}$  astfel:

$$A = G^T R$$

unde:

$$G = G_{n-1,m} G_{n-2,m} G_{n-2,m-1} \dots G_{1m} \dots G_{13} G_{12}, \quad R = GA.$$

O matrice de rotație Givens, notată  $G_{kl}$  este folosită pentru a elimina (a anula) elementul  $A(l, k)$  de sub diagonala principală ( $k < l$ ) și are forma:

$$G_{kl} = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \cos \theta & \cdots & -\sin \theta & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sin \theta & \cdots & \cos \theta & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

Pentru a determina matricea de rotație Givens  $G_{kl}$ , vom folosi relațiile:

$$\rho = \sqrt{A(k, k)^2 + A(l, k)^2}$$

$$s = \sin \theta = -\frac{A(l, k)}{\rho}$$

$$c = \cos \theta = \frac{A(k, k)}{\rho}$$

Datorită formei matricelor de rotație Givens, înmulțirea  $G_{kl}x$ , unde  $x$  este vector coloană, se efectuează astfel:

- $x(k) = c \cdot x(k) - s \cdot x(l)$ ;
- $x(l) = s \cdot x(k) + c \cdot x(l)$ ;
- restul elementelor rămân neschimbate.

## Algoritmul Gram-Schmidt

Vom considera relația  $A = QR$ , adică:

$$\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

cu necunoscutele  $q_i$  și  $r_{ij}$ , ( $i \leq j$ ), unde:

- $a_i$  este coloana  $i$  din matricea  $A$ ;
- $q_i$  este coloana  $i$  din matricea  $Q$ ;
- $r_{ij}, i \leq j$ , reprezintă elementele din matricea superior triunghiulară  $R$ .

Algoritmul Gram-Schmidt este schițat în continuare:

Pentru  $j = 1 : n$

$$r_{ij} = q_i^T a_j, \quad i = 1 : j - 1;$$

$$aux = a_j - \sum_{i=1}^{j-1} r_{ij} q_i;$$

$$r_{jj} = \|aux\|_2;$$

$$q_j = \frac{aux}{r_{jj}}.$$

## Polinoame ortogonale

Un polinom ortogonal este definit prin:

- relația de recurență;
- cazurile de bază ale relației de recurență;
- intervalul  $[(a, b)]$  pe care este definit;
- funcția pondere  $w(x)$ , folosită la calcularea produsului scalar.

Exemple de polinoame ortogonale:

- *Cebâșev*:

$$T_{n+1} - 2xT_n + T_{n-1} = 0, \quad T_0 = 1, \quad T_1 = x;$$

$$(-1, 1); \quad w(x) = \frac{1}{\sqrt{1-x^2}};$$

- *Legendre*:

$$(n+1)L_{n+1} - (2n+1)xL_n + nL_{n-1} = 0, \quad L_0 = 1, \quad L_1 = x;$$

$$[-1, 1]; \quad w(x) = 1;$$

- *Laguerre*:

$$G_{n+1} - (2n+1-x)G_n + n^2G_{n-1} = 0, \quad G_0 = 1, \quad G_1 = 1-x;$$

$$[0, \infty); \quad w(x) = e^{-x};$$

- *Hermite*:

$$H_{n+1} - 2xH_n + 2nH_{n-1} = 0, \quad H_0 = 1, \quad H_1 = 2x;$$

$$(-\infty, \infty); \quad w(x) = e^{-x^2}$$

Proprietățile polinoamelor ortogonale sunt:

1. Orice polinom ortogonal are radacinile în intervalul  $[(a, b)]$ , reale și distincte;
2. Orice polinom ortogonal este ortogonal cu orice polinom de grad mai mic decât el.

Polinoamele  $p_0, p_1, \dots, p_n$  reprezintă o bază de polinoame ortogonale, dacă:

- $\|p_i\| = 1, \quad \forall i$ ; și
- $\langle p_i, p_j \rangle = 0, \quad \forall i \neq j$ , unde  $\langle p_i, p_j \rangle = \int_a^b p_i(x)p_j(x)w(x)dx$ .

Operații utile cu polinoame în Octave:

```
1 function test_poly
2     %%% Coeficientii polinomului p
3     p = [ 2 1 -1]
4     %%% Afisarea polinomului
5     polyout(p, 'x')
6     %%% Coeficientii polinomului q
7     q = [ 1 2]
8     %%% Afisarea polinomului
9     polyout(q, 'x')
10    %%% Produsul dintre p si q
11    r = conv(p,q)
12    %%% Afisarea rezultatului
13    polyout(r, 'x')
14 endfunction
```

Listing 1: Polinoame în Octave.

## Probleme rezolvate

### Problema 1

Dacă  $H_1, H_2$  sunt matrici ortogonale, arătați că produsul  $H_1 H_2$  este o matrice ortogonală.

*Soluție:*

Din proprietățile matricelor ortogonale, avem:  $H$  ortogonală  $\Rightarrow HH^T = I_n$  și

$$\begin{cases} H_1 H_1^T = I_n \\ H_2 H_2^T = I_n \end{cases} \Rightarrow (H_1 H_2) (H_1 H_2)^T = H_1 H_2 H_2^T H_1^T = I_n$$

## Problema 2

Fie un reflector elementar Householder

$$H = I_n - \frac{2uu^T}{\|u\|^2}$$

a) Arătați că  $H^T = H$ ;

b) Calculați  $Hu$ .

*Soluție:*

$$\text{a) } H^T = \left( I_n - \frac{2uu^T}{\|u\|^2} \right)^T = I_n - \frac{2uu^T}{\|u\|^2} = H$$

$$\text{b) } Hu = \left( I_n - \frac{2uu^T}{\|u\|^2} \right) u = u - \frac{2uu^T}{\|u\|^2} u = u - 2u = -u$$

## Problema 3

Determinați factorizarea QR folosind transformarea Householder pentru matricea:

$$A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}.$$

*Soluție:*

Prima iterație:

$$a_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}, \quad \sigma_1 = \|a_1\|_2 = 3 \quad u_1 = \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \quad \|u_1\|_2^2 = 6;$$

---

$$H_1 = I_3 - \frac{2u_1u_1^T}{\|u_1\|_2^2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3} \begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{bmatrix}$$

$$A_2 = H_1 A_1 = \frac{1}{3} \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & -2 \\ 2 & -2 & -1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 3 \\ 0 & 3 & 3 \end{bmatrix}$$

A doua iterație:

$$\bar{a}_2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \quad \sigma_2 = \|\bar{a}_2\|_2 = 3 \quad u_2 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \quad \|u_2\|_2^2 = 2;$$

$$H_2 = I_3 - \frac{2u_2u_2^T}{\|u_2\|_2^2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$A_3 = H_2 A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 3 \\ 0 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 3 & 3 \\ 0 & 0 & 3 \end{bmatrix}$$

Atunci:

$$R = A_3 = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 3 & 3 \\ 0 & 0 & 3 \end{bmatrix}$$

$$Q = (H_2 H_1)^T = H_1 H_2 = \frac{1}{3} \begin{bmatrix} 2 & 2 & 1 \\ 1 & -2 & 2 \\ 2 & -1 & -2 \end{bmatrix}$$

#### Problema 4

Să se scrie un program OCTAVE care să implementeze transformarea Givens. Programul primește ca date de intrare:  $A$  - matricea sistemului;  $b$  - vectorul termenilor liberi. Rezultatele programului vor fi:  $Q$  - matricea factor ortogonală;  $R$  - matricea factor superior triunghiular;  $\tilde{b}$  - vectorul modificat al termenilor liberi.



```
1 function [Q R b] = givens(A, b)
2   [m n] = size(A);
3   G = eye(m);
4
5   for k = 1 : n-1
6     for l = k + 1 : m
7       r = sqrt(A(k,k)^2 + A(l,k)^2);
8       c = A(k,k)/r;
9       s = -A(l,k)/r;
10
11      t = c*A(k,k:n) - s*A(l,k:n);
12      A(l,k:n) = s*A(k,k:n) + c*A(l,k:n);
13      A(k,k:n) = t;
14
15      u = c*b(k) - s*b(l);
16      b(l) = s*b(k) + c*b(l);
17      b(k) = u;
18
19      t = c*G(k,1:m) - s*G(l,1:m);
20      G(l,1:m) = s*G(k, 1:m) + c*G(l,1:m);
21      G(k,1:m) = t;
22    end
23  end
24
25  Q = G';
26  R = A;
27 end
```

Listing 2: Transformarea Givens.

## Problema 5

Să se scrie un program OCTAVE pentru calculul coeficienților unui polinom ortogonal de grad  $n$ . Selecția polinomului se face printr-un parametru șir de caractere care poate avea valorile: ‘cebasev’, ‘legendre’, ‘laguerre’ sau ‘hermite’.

*Soluție:*

```
1 function P = poliOrtogonal(num_e_polinom, n)
```

```
2
3  if strcmp(ume_polinom, 'legendre')
4      P0 = [1];
5      P1 = [1 0];
6      C0 = @(n) ([-n/(n+1)]);
7      C1 = @(n) ([ (2*n+1)/(n+1) 0]);
8  endif
9
10 if strcmp(ume_polinom, 'cebasev')
11     P0 = [1];
12     P1 = [1 0];
13     C0 = @(n) ([-1]);
14     C1 = @(n) ([2 0]);
15 endif
16
17 if strcmp(ume_polinom, 'laguerre')
18     P0 = [1];
19     P1 = [-1 1];
20     C0 = @(n) ([-n^2]);
21     C1 = @(n) ([-1 2*n+1]);
22 endif
23
24 if strcmp(ume_polinom, 'hermite')
25     P0 = [1];
26     P1 = [2 0];
27     C0 = @(n) ([-2*n]);
28     C1 = @(n) ([2 0]);
29 endif
30
31 P = poliOrtogonalGeneral(n, P0, P1, C0, C1);
32 endfunction
```

```
1 function P = poliOrtogonalGeneral(n, P0, P1, C1, C2)
2     %Pn+1 = C1(n)*Pn + C2(n)*Pn-1
3     %Valori initiale: P0, P1
4     %Iesire: Pn
5
6     %Cazuri de baza
7     if n == 0
8         P = P0;
```

```
9   endif
10
11   if n == 1
12       P = P1;
13   endif
14
15   %Ultimii 2 termeni ai recurentei sunt pastrati in P0 si P1
16   for i=2:n
17       Paux1 = conv(P0, C1(i));
18       Paux2 = conv(P1, C2(i));
19
20       m = length(Paux1);
21       n = length(Paux2);
22       r = max(m, n);
23
24       Paux1(r-m+1:r) = Paux1(1:m);
25       Paux1(1:r-m) = zeros(1, r-m);
26
27       Paux2(r-n+1:r) = Paux2(1:n);
28       Paux2(1:r-n) = zeros(1, r-n);
29
30       Paux = Paux1 + Paux2;
31
32       P0 = P1;
33       P1 = Paux;
34   endfor
35
36   P = P1;
37 endfunction
```

## Probleme propuse

### Problema 1

Se consideră vectorii  $u, v \in \mathbb{R}^n$  ortonormați ( $\|u\|_2 = 1$ ,  $\|v\|_2 = 1$ ,  $u^T v = v^T u = 0$ ).  
Se formează vectorul  $x = u + v$ .

- a) Să se dea exemplu de doi vectori ortonormați;

- b) Să se calculeze  $\|x\|_2$ ;
- c) Se formează matricea  $H = I_n - xx^T$ . Să se calculeze  $Hu$ ,  $Hv$  și  $\|H\|_2$ ;
- d) Dacă  $A = uv^T$ , calculați  $B = H^{-n}AH^n$ .

## Problema 2

Implementați transformarea Householder pentru o matrice  $A$ , astfel:

- a) Implementați o funcție care primește un vector  $x$ , un index  $p$  și calculează parametrii  $\sigma, v_p, \beta$  definiți mai sus:

```
function [vp, sigma, beta] = GetHSReflector(x, p)
```

- b) Implementați o funcție care primește un vector  $x$ , un index  $p$ , parametrul  $\sigma$  și calculează transformarea Householder aplicată asupra vectorului, considerând că acest vector a fost folosit la calculul parametrilor (coloana  $p$  din  $A$ ).

```
function x = ApplyHSToPColumn(x, p, sigma)
```

- c) Implementați o funcție care primește un vector oarecare  $x$ , un index  $p$ , vectorul Householder  $v_p$  parametrul  $\beta$  și calculează transformarea Householder aplicată asupra vectorului (coloanele  $p+1:n$  din  $A$ ).

```
function x = ApplyHSToRandomColumn(x, vp, p, beta)
```

- d) Implementați funcția

```
function [Q, R] = Householder(A),
```

folosind funcțiile definite mai sus.

## Problema 3

Să se determine descompunerea QR pentru matricea  $A = \begin{bmatrix} 3 & 1 & -2 \\ 1 & 3 & 1 \\ -2 & 1 & 3 \end{bmatrix}$  folosind transformarea Givens.

### **Problema 4**

Să se scrie un program OCTAVE pentru a implementa algoritmul Gram-Schmidt.