Laborator 6

Descompunerea valorilor singulare

6.1 Preliminarii

6.1.1 Descompunerea valorilor singulare

Vom introduce descompunerea valorilor singulare (DVS) ale unei matrice prin următoarea teoremă.

Teorema 6.1 *Pentru orice matrice $A \in \mathbf{R}^{m\times n}$ există matricele ortogonale $U \in \mathbf{R}^{m\times m}$ şi $V \in \mathbf{R}^{n\times n}$ astfel încât*

$$U^T AV = \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.1}$$

unde

$$\Sigma_1 = \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_r) \in \mathbf{R}^{r\times r}, \tag{6.2}$$

cu

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0. \tag{6.3}$$

Expresiile (6.1)-(6.3) definesc descompunerea valorilor singulare ale matricei A.

Numerele pozitive σ_i, $i = 1 : p, p = \min(m, n)$, ordonate descrescător

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,\ \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0 \tag{6.4}$$

se numesc valori singulare *ale matricei A. Coloanele $u_j \in \mathbf{R}^m$ ale matricei ortogonale U se numesc* vectori singulari *la stânga ale matricei A. Coloanele $v_j \in \mathbf{R}^n$ ale matricei ortogonale V se numesc* vectori singulari *(la dreapta) ale matricei A.*

DVS a unei matrice scoate în evidenţă numeroase aspecte structurale ale matricei respective şi are multiple valenţe aplicative. Câteva din cele mai importante aplicaţii sunt prezentate succint în paragraful următor.

Calculul DVS are la bază următorul rezultat.

Teorema 6.2 *Valorile singulare nenule $\sigma_i,\ i = 1 : r$, ale unei matrice $A \in \mathbf{R}^{m\times n}$ sunt rădăcinile pătrate pozitive ale valorilor proprii nenule ale matricei simetrice pozitiv semidefinite*

$$B = A^T A \tag{6.5}$$

i.e., dacă $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$, sunt cele r valori proprii nenule ale lui B atunci

$$\sigma_i = \sqrt{\lambda_i},\ i = 1 : r. \tag{6.6}$$

Teorema 6.2 sugerează o procedură pentru calculul valorilor singulare ale unei matrice A date folosind algoritmul QR simetric pentru calculul valorilor proprii ale matricei $B = A^T A$. Aceasta procedură nu este recomandată datorită posibilei rele condiţionări numerice a matricei B. G.H.Golub şi W.Kahan au elaborat un algoritm (prezentat în secţiunea următoare) care evită calculul explicit al lui B.

6.1.2 Aplicaţii ale DVS

Prezentăm succint câteva aplicaţii ale DVS. Pentru detalii şi alte aplicaţii recomandăm consultarea cursului.

Calculul rangului unei matrice

Fie $A = U\Sigma V^T$ DVS a matricei $A \in \mathbf{R}^{m\times n}$. Întrucât înmulţirea cu matrice nesingulare nu modifică rangul unei matrice, rangul lui A este rangul lui Σ. Prin urmare, rangul unei matrice este dat de numărul valorilor sale singulare nenule.

În general, din cauza erorilor de rotunjire, valorile proprii *calculate* $\hat{\sigma}_i$ vor fi toate nenule, i.e.

$$\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_p > 0. \tag{6.7}$$

De aceea în practica numerică se utilizează conceptul de *rang numeric*. Rangul numeric se obţine neglijând valorile singulare mici, e.g. inferioare unei toleranţe precizate ε

$$\hat{\sigma}_i \leq \varepsilon. \tag{6.8}$$

Dacă indicele $i = r + 1$ este primul pentru care (6.8) este satisfăcută, atunci *rangul numeric* al matricei A este

$$\text{rang}(A, \varepsilon) = r. \tag{6.9}$$

Rezolvarea problemei generale CMMP

Problema generală CMMP constă în rezolvarea sistemelor liniare (determinate, supradeterminate sau subdeterminate)

$$Ax = b \tag{6.10}$$

cu matricea $A \in \mathbf{R}^{m\times n}$ de rang posibil nemaximal. În general, în cazul

$$\text{rang}A < \min(m,n) \tag{6.11}$$

sistemul liniar (6.10) nu are soluţii dar are un număr infinit de (pseudo)soluţii în sens CMMP. Problema este calculul (pseudo)soluţiei de normă euclidiană minimă

$$\|x^*\| = \min_{\min_{x\in\mathbf{R}^n}\|b-Ax\|} \|x\|. \tag{6.12}$$

Următoarea teoremă arată cum se calculează *(pseudo)soluţia normală* utilizând DVS.

Teorema 6.3 *Fie $A \in \mathbf{R}^{m\times n}$ cu* rang$A = r \leq \min(m,n)$. *Dacă $U^TAV = \Sigma$ este DVS a lui A atunci pseudosoluţia de normă euclidiană minimă pentru (6.10) este dată de*

$$x^* = \sum_{j=1}^{r} \frac{u_j^T b}{\sigma_j} v_j \tag{6.13}$$

unde $u_j = Ue_j = U(:,j) \in \mathbf{R}^m$ şi $v_j = Ve_j = V(:,j) \in \mathbf{R}^n$ sunt coloanele j ale lui U şi, respectiv, V. Mai mult, reziduul minim este

$$(\rho^*)^2 = \|Ax^* - b\|_2^2 = \sum_{j=r+1}^{m} (u_j^T b)^2. \tag{6.14}$$

Schema de calcul a pseudosoluţiei normale este directă şi evidentă.

Calculul bazelor ortogonale pentru subspaţii liniare

DVS a matricei $A \in \mathbf{R}^{m\times n}$ produce baze ortogonale pentru toate subspaţiile liniare definite de A. Concret avem

$$\begin{array}{ll} (a) & \mathcal{S}_1 = \text{Im}A = \text{Im}U(:,1:r) \subset \mathbf{R}^m \\ (b) & \mathcal{T}_1 = \text{Ker}A = \text{Im}V(:,r+1:n) \subset \mathbf{R}^n \\ (c) & \mathcal{S}_2 = \text{Ker}A^T = \text{Im}U(:,r+1:m) \subset \mathbf{R}^m \\ (d) & \mathcal{T}_2 = \text{Im}A^T = \text{Im}V(:,1:r) \subset \mathbf{R}^n, \end{array} \tag{6.15}$$

i.e. *primele r coloane ale matricei U formează o bază ortogonală pentru subspaţiul liniar* ImA, *ultimele $n-r$ coloane ale matricei V formează o bază ortogonală pentru subspaţiul liniar* Ker$A$ etc.

Printre alte aplicaţii ale DVS menţionăm diverse operaţii cu subspaţii liniare, calculul pseudoinversei, rezolvarea unor probleme de optimizare cu şi fără restricţii etc.

6.2 Algoritmul DVS

Cea mai bună metodă pentru calculul DVS a fost propusă de *G.H.Golub şi W.Kahan* în 1965. Algoritmul propus de ei determină simultan U şi V aplicând într-un mod implicit şi indirect algoritmul QR simetric matricei $B = A^T A$ (i.e. acţionând numai asupra matricei A fără a forma explicit matricea B).

Algoritmul DVS se desfăşoară în două etape.

1. *Prima etapă* constă în reducerea lui A la o formă bidiagonală J astfel încât matricea tridiagonală $T = J^T J$ coincide cu cea produsă de prima etapă a algoritmului QR simetric aplicat lui B.

2. *Etapa a doua* constă în reducerea iterativă a matricei bidiagonale J la forma diagonală prin anularea asimptotică a elementelor supradiagonale. Aceasta poate fi obţinută folosind transformări ortogonale bilaterale ce corespund unui pas QR simetric cu deplasare implicită aplicat lui B.

6.2.1 Etapa 1: bidiagonalizarea

Fie $A \in \mathbf{R}^{m\times n}$ şi, pentru precizare, $m > n$. Vom arăta cum se calculează matricele ortogonale $U \in \mathbf{R}^{m\times m}$ şi $V \in \mathbf{R}^{n\times n}$ astfel încât

$$J = U^T A V = \begin{bmatrix} f_1 & g_1 & & & \\ & f_2 & g_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & g_{n-1} \\ & & & & f_n \\ & & & & \\ & & & & \end{bmatrix} \tag{6.16}$$

este bidiagonală. Matricele U şi V vor fi calculate ca produse de transformări Householder (reflectori elementari)

$$U = U_1 U_2 \cdots U_n, \qquad V = V_2 V_3 \cdots V_{n-1} \tag{6.17}$$

folosind următoarea schemă de calcul

1. $U \leftarrow I_m$
2. $V \leftarrow I_n$
3. **Pentru** $k = 1 : n$
   1. Se calculează reflectorul U_k astfel încât $(U_k^T A)(k+1 : m, k) = 0$
   2. $A \leftarrow U_k^T A$
   3. $U \leftarrow U U_k$
   4. **Dacă** $k < n - 1$
      1. Se calculează reflectorul V_{k+1} astfel încât $(A V_{k+1})(k, k+2 : n) = 0$
      2. $A \leftarrow A V_k$
      3. $V \leftarrow V V_k$

De notat faptul că la pasul k reflectorul U_k are structura $U_k = \mathrm{diag}(I_{k-1}, \overline{U}_k)$ şi, prin urmare, zerourile create în primele $k-1$ linii şi primele $k-1$ coloane sunt conservate. Similar, $V_{k+1} = \mathrm{diag}(I_k, \overline{V}_{k+1})$ nu afectează zerourile create în primele k coloane şi primele $k-1$ linii. În final, matricea A este suprascrisă de matricea bidiagonală

$$A \leftarrow U_n^T \cdots U_2^T U_1^T A V_2 V_3 \cdots V_{n-1} = U^T A V. \tag{6.18}$$

Algoritmul detaliat corespunzător schemei de mai sus este următorul.

ALGORITM 6.4 *JQ (Bidiagonalizarea)* Dată matricea $A \in \mathbf{R}^{m\times n}$ cu $m > n$, algoritmul calculează matricele ortogonale $U \in \mathbf{R}^{m\times m}$ şi $V \in \mathbf{R}^{n\times n}$ şi matricea bidiagonală J astfel încât $J = U^T A V$. Matricea J suprascrie matricea A dar se reţin numai elementele diagonale ale lui J în vectorul $f \in \mathbf{R}^n$ şi cele supradiagonale în vectorul $g \in \mathbf{R}^{n-1}$. Se folosesc vectorii de lucru u şi v, precum şi scalarii β şi γ, pentru definirea reflectorilor curenţi $U_k = I_m - uu^T/\beta$ şi $V_{k+1} = I_n - vv^T/\gamma$.

1. $U = I_m$
2. $V = I_n$
3. **Pentru** $k = 1 : n$

% Calculează $U_k = U_k^T$, $A \leftarrow U_k A$ şi $U = UU_k$

1. $\sigma = \mathrm{sign}(A(k,k))\sqrt{\sum_{i=k}^{m} A(i,k)^2}$
2. $u(k) = A(k,k) + \sigma$
3. **Pentru** $i = k+1 : m$
   1. $u(i) = A(i,k)$
4. $\beta = u(k)\sigma$
5. $A(k,k) \leftarrow -\sigma$

% $A \leftarrow U_k A$

6. **Pentru** $i = k+1 : m$
   1. $A(i,k) \leftarrow 0$
7. **Dacă** $k < n$
   1. **Pentru** $j = k+1 : n$
      1. $\tau = (\sum_{i=k}^{m} u(i)A(i,j))/\beta$
      2. **Pentru** $i = k : m$
         1. $A(i,j) \leftarrow A(i,j) - \tau u(i)$

% $U \leftarrow UU_k$

8. **Pentru** $i = 1 : m$
   1. $\tau = (\sum_{j=k}^{m} U(i,j)u(j))/\beta$
   2. **Pentru** $j = k : m$
      1. $U(i,j) \leftarrow U(i,j) - \tau u(j)$

% Calculează V_{k+1}, $A \leftarrow AV_{k+1}$ şi $V \leftarrow VV_{k+1}$

9. **Dacă** $k < n-1$

% V_{k+1}

   1. $\sigma = \mathrm{sign}(A(k,k+1))\sqrt{\sum_{j=k+1}^{n} A(k,j)^2}$
   2. $v(k+1) = A(k,k+1) + \sigma$

3. **Pentru** $j = k+2 : n$
   1. $v(j) = A(k,j)$
4. $\gamma = v(k+1)\sigma$

% $A \leftarrow AV_{k+1}$

5. $A(k,k+1) \leftarrow -\sigma$
6. **Pentru** $j = k+2 : n$
   1. $A(k,j) \leftarrow 0$
7. **Pentru** $i = k+1 : m$
   1. $\tau = (\sum_{j=k+1}^{n} A(i,j)v(j))/\gamma$
   2. **Pentru** $j = k+1 : n$
      1. $A(i,j) \leftarrow A(i,j) - \tau v(j)$

% $V \leftarrow VV_{k+1}$

8. **Pentru** $i = 1 : n$
   1. $\tau = (\sum_{j=k+1}^{n} V(i,j)v(j))/\gamma$
   2. **Pentru** $j = k+1 : n$
      1. $V(i,j) \leftarrow V(i,j) - \tau v(j)$

4. **Pentru** $i = 1 : n-1$
   1. $f(i) = A(i,i)$
   2. $g(i) = A(i,i+1)$
5. $f(n) = A(n,n)$

Recomandăm scrierea unui program distinct de bidiagonalizare având ca date de intrare matricea A şi ca ieşiri vectorii $f \in \mathbf{R}^n$ al elementelor diagonale ale matricei bidiagonale J şi $g \in \mathbf{R}^{n-1}$ al elementelor supradiagonale ale matricei bidiagonale J, precum şi matricele de transformare U şi V, conform sintaxei

$$[f, g, U, V] = \mathbf{JQ}(A).$$

6.2.2 Etapa 2: diagonalizarea iterativă

Diagonalizarea iterativă produce un şir de matrice

$$J_1 = J, J_2, \cdots, J_k, \cdots \tag{6.19}$$

convergent către matricea diagonală $\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}$ astfel încât şirul de matrice

$$T_1 = J_1^T J_1, \cdots, T_k = J_k^T J_k, \cdots \tag{6.20}$$

este şirul QR simetric cu deplasare implicită convergent către $\begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbf{R}^{n\times n}$, care este o formă Schur a matricei tridiagonale $T = T_1$.

Presupunem că matricea $J_k := J$ are forma bidiagonală (6.16) şi aplicăm un pas QR cu deplasare implicită matricei tridiagonale

$$T = T_k = J_k^T J_k = J^T J. \tag{6.21}$$

Un pas QR simetric, cu deplasare implicită, pentru matricea tridiagonală T constă din:

1. Calculul deplasării

$$\mu = T(n,n) = g_{n-1}^2 + f_n^2 \tag{6.22}$$

sau, şi mai bine, a *deplasării Wilkinson*, care este valoarea proprie cea mai apropiată de $g_{n-1}^2 + f_n^2$ a matricei

$$T(n-1:n, n-1:n) = \begin{bmatrix} g_{n-2}^2 + f_{n-1}^2 & f_{n-1}g_{n-1} \\ f_{n-1}g_{n-1} & g_{n-1}^2 + f_n^2 \end{bmatrix}. \tag{6.23}$$

2. Calculul unei matrice ortogonale U_1 astfel încât $U_1 e_1$ este prima coloană a matricei de transformare de la pasul QR curent cu deplasare explicită

$$U_1 e_1 = \rho \begin{bmatrix} t_{11} - \mu \\ t_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \rho \begin{bmatrix} f_1^2 - \mu \\ g_1 f_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{6.24}$$

Matricea U_1 poate fi o rotaţie Givens P_{12} astfel încât

$$P_{12}^T \begin{bmatrix} f_1^2 - \mu \\ g_1 f_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{\rho} e_1. \tag{6.25}$$

3. Calculul matricei

$$C = P_{12}^T T P_{12} \tag{6.26}$$

punându-se în evidenţă o alterare a structurii tridiagonale în poziţia $(3,2)$.

4. Aplicarea algoritmului HQ (adaptat adecvat pentru cazul simetric, i.e. *algoritmul de tridiagonalizare*) matricei C. Se obţine matricea

$$T = Q^T C Q \tag{6.27}$$

în formă tridiagonală. Matricea Q poate fi o secvenţă de rotaţii:

$$Q = P_{23} \cdots P_{n-1,n}, \tag{6.28}$$

astfel încât noua matrice este

$$T \leftarrow T' = Q^T C Q = Q^T P_{12}^T T P_{12} Q = P_{n-1,n}^T \cdots P_{23}^T P_{12}^T T P_{12} P_{23} \cdots P_{n-1,n}. \tag{6.29}$$

Ideea de bază a unui pas DVS constă în a acţiona printr-o transformare ortogonala bilaterală $J' = U^T JV$ astfel încât $J'^T J' = T'$. Această idee a unui pas DVS Golub-Kahan este detaliată în cele ce urmează.

1. *Se aplică rotaţia Givens P_{12}, de mai sus, matricei J în loc de matricea T*

$$K = JP_{12} \tag{6.30}$$

ceea ce alterează structura bidiagonală a matricei J în poziţia $(1, 2)$.

2. *Se readuce structura alterată la forma bidiagonală* prin transformări ortogonale bilaterale.

$$J \leftarrow J' = U_{n-1}U_{n-2}\cdots U_1 K V_2 \cdots V_{n-1} \tag{6.31}$$

unde U_s, V_s pot fi rotaţii Givens (sau reflectori Householder).

Schema de calcul pentru (6.31) este următoarea.

Pentru $k = 1 : n-1$
1. Se calculează rotaţia Givens $U_k := P_{k,k+1}$ astfel încât $(U_k^T K)(k+1, k) = 0$
2. $K \leftarrow U_k^T K$
3. $U \leftarrow UU_k$
4. **Dacă** $k < n-1$
   1. Se calculează rotaţia Givens $V_{k+1} := P_{k+1,k+2}$ astfel încât $(KV_{k+1})(k, k+2) = 0$
   2. $K \leftarrow KV_{k+1}$
   3. $V \leftarrow VV_{k+1}$

Noua matrice $K = J'$ este bidiagonală şi

$$T' = J'^T J' = (U_{n-1}^T \cdots U_1^T JP_{12}V_2 \cdots V_{n-1})^T * U_{n-1}^T \cdots U_1^T JP_{12}V_2 \cdots V_{n-1} =$$

$$= V_{n-1}^T \cdots V_2^T P_{12}^T J^T U_1 \cdots U_{n-1}U_{n-1}^T \cdots U_1^T JP_{12}V_2 \cdots V_{n-1} = Q^T J^T JQ \tag{6.32}$$

şi prima coloană a matricei de transformare

$$Qe_1 = P_{12}V_2 \cdots V_{n-1}e_1 = P_{12}e_1$$

este aceeaşi ca în pasul QR simetric pentru matricea tridiagonală T.

În consecinţă, matricea $J_k = J$, definind şirul DVS este astfel încât $T_k = J_k^T J_k$ este şirul QR pentru matricea $B = A^T A$ şi, prin urmare, este convergentă la o formă diagonală.

Este posibil ca matricea diagonală limită $\Sigma = J_\infty$ să nu aibă elementele diagonale ordonate. Ordonarea poate fi obţinută imediat cu o secvenţă de permutări şi este lăsată în sarcina studentului.

ALGORITM 6.5 *(Iteraţie (pas) DVS)* Daţi vectorii $f \in \mathbf{R}^n$ şi $g \in \mathbf{R}^{n-1}$ ce definesc matricea bidiagonală J din (6.16), algoritmul calculează vectorii f' şi g' ce definesc matricea succesor $J' = U^T JV$ din şirul DVS. Matricea J' suprascrie matricea J. Mai precis, noii vectori f' şi g' suprascriu vectorii f şi g.

1. % Se calculează deplasarea (pentru $n > 2$ deplasarea Wilkinson)
   Dacă $n > 2$
      1. $\delta = (g_{n-2}^2 + f_{n-1}^2 - g_{n-1}^2 - f_n^2)/2$
      2. $\eta = (g_{n-1} f_{n-1})^2$
      3. $\mu = g_{n-1}^2 + f_n^2 + \dfrac{\eta}{\delta + \text{sign}(\delta)\sqrt{\delta^2 + \eta}}$

   altfel
      4. $\mu = g_{n-1}^2 + f_n^2$
2. $y = f_1^2 - \mu$; $z = g_1 f_1$
3. % Se calculează c, s astfel încât $P_{12}^T \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$
   1. $r = \sqrt{y^2 + z^2}$
   2. $c = y/r$
   3. $s = -z/r$
4. % Se calculează $J \leftarrow JP_{12}$. Fie χ elementul nenul
   % alterant al structurii bidiagonale.
   1. $\alpha = f_1 c - g_1 s$
   2. $g_1 = f_1 s + g_1 c$
   3. $f_1 = \alpha$
   4. $\chi = -f_2 s$
   5. $f_2 = f_2 c$
5. % Se calculează $V \leftarrow VP_{12}$.
   Pentru $i = 1 : n_V$
      1. $\alpha = v_{i1} c - v_{i2} s$
      2. $v_{i2} = v_{i1} s + v_{i2} c$
      3. $v_{i1} = \alpha$
6. % Readucerea la forma bidiagonală (6.31)
   Pentru $k = 1 : n - 1$
      1. $r = \sqrt{f_k^2 + \chi^2}$
      2. $c = f_k / r$
      3. $s = -\chi / r$
      4. $f_k = r$
      5. $\alpha = c g_k - s f_{k+1}$
      6. $f_{k+1} = s g_k + c f_{k+1}$
      7. $g_k = \alpha$
      8. **Dacă** $k < n - 1$
         1. $\chi = -s g_{k+1}$
         2. $g_{k+1} = c g_{k+1}$
      9. % Se calculează $U \leftarrow UU_k$.
      Pentru $i = 1 : m_U$
         1. $\alpha = u_{i,k} c - u_{i,k+1} s$
         2. $u_{i,k+1} = u_{i,k} s + u_{i,k+1} c$
         3. $u_{i,k} = \alpha$

10. **Dacă** $k < n-1$
    1. $r = \sqrt{g_k^2 + \chi^2}$
    2. $c = g_k / r$
    3. $s = -\chi / r$
    4. $g_k = r$
    5. $\alpha = f_{k+1} c - g_{k+1} s$
    6. $g_{k+1} = f_{k+1} s + g_{k+1} c$
    7. $f_{k+1} = \alpha$
    8. $\chi = -f_{k+2} s$
    9. $f_{k+2} = f_{k+2} c$
    10. % Se calculează $V \leftarrow V V_{k+1}$.

    Pentru $i = 1 : n_V$
        1. $\alpha = v_{i,k+1} c - v_{i,k+2} s$
        2. $v_{i,k+2} = v_{i,k+1} s + v_{i,k+2} c$
        3. $v_{i,k+1} = \alpha$

După cum se observă algoritmul face şi actualizarea matricelor de transformare. Întrucât, pe parcursul iteraţiilor DVS, se va acţiona asupra unor vectori f şi g de lungime variabilă (din ce în ce mai mici) iar matricele de transformare care se actualizează pastrează ordinul iniţial, în algoritm acest ordin este notat cu m_U, respectiv n_V.

Vom introduce pentru algoritmul de mai sus sintaxa

$$[f, g, U, V] = \mathbf{pas\_DVS}(f, g, U, V).$$

Tipic, după caţiva paşi DVS g_{n-1} devine neglijabil şi dimensiunea problemei scade cu o unitate. Pentru a controla procesul de anulare asimptotică a elementelor se foloseşte următorul criteriu

$$\text{daca } \ |g_i| \leq \text{tol}(|f_i| + |f_{i+1}|) \text{ atunci } g_i = 0, \tag{6.33}$$

unde tol este de ordinul de mărime al erorilor de rotunjire.

Faza iterativă a algoritmului DVS constă în aplicarea iterativă a pasului DVS părţii ireductibile a matricei bidiagonale J şi, simultan, monitorizarea anulării elementelor supradiagonale şi diagonale. În final se obţine o matricea diagonală ale cărei elemente diagonale se ordonează prin permutări simultane de linii şi coloane (i.e. permutări diagonale). Matricea diagonală ordonată, împreună cu matricele de transformare, definesc descompunerea valorilor singulare.

Prezentăm mai jos o versiune simplificată a algoritmului DVS, cu o monitorizare mai simplă a elementelor care se anulează. Anularea are loc efectiv dacă modulul unui element supradiagonal devine inferior unei toleranţe date, conform criteriului (6.33). Concret această versiune ţine seama de faptul că, în general, anularea elementelor supradiagonale are loc oarecum ordonat, începând de la ultimul element şi mergând spre primul.

Pentru mai multă claritate algoritmul este prevăzut cu comentarii care pun în evidenţă explicaţiile de mai sus.

ALGORITM 6.6 *(Algoritmmul DVS)* Date o matrice $A \in \mathbf{R}^{m\times n}$, cu $m > n$, şi nivelul de toleranţă tol, algoritmul calculează matricea diagonală $\Sigma \in \mathbf{R}^{m\times n}$ şi matricele ortogonale $U \in \mathbf{R}^{m\times m}$, $V \in \mathbf{R}^{n\times n}$ astfel încât $\Sigma = U^T AV$.

1. % Bidiagonalizarea
   $[f, g, U, V] = \mathbf{JQ}(A)$
2. $q = n$
3. **Cât timp** $q > 1$
   1. % Iteraţia DVS
      $[f(1:q), g(1:q-1), U, V] = \mathbf{pas\_DVS}(f(1:q), g(1:q-1), U, V)$
   2. % Anularea elementelor neglijabile
      1. index = 1
      2. **Cât timp** index = 1
         1. **Dacă** $|g_{q-1}| \leq \text{tol}(|f_{q-1}| + |f_q|)$
            1. $g_{q-1} = 0$
            2. $q = q - 1$

            altfel
            3. index = 0
         2. **Dacă** $q = 1$
            1. index = 0
4. % Valorile singulare trebuie să fie pozitive
   1. **Pentru** $i = 1 : n$
      1. **Dacă** $f(i) < 0$
         1. $f(i) = -f(i)$
         2. $V(i,:) = -V(i,:)$
5. $\Sigma = \text{diag}(f)$
6. $\Sigma = [\Sigma; \text{zeros}(m-n, n)]$.

Se demonstrează că algoritmul de mai sus este numeric stabil în sensul că descompunerea valorilor singulare calculată este descompunerea valorilor singulare exactă a unei matrice ce diferă nesemnificativ de matricea iniţială dată.

6.3 Sarcini de lucru

6.3.1 A. In laborator

1. Se va edita şi testa un program MATLAB pentru implementarea algoritmului JQ de bidiagonalizare a unei matrice A prin transformări ortogonale bilaterale. Programul va afişa matricea A curentă pentru vizualizarea procesului de bidiagonalizare. Testarea se va realiza prin calculul normei reziduului $A - UJV^T$.

2. Se va edita şi testa un program MATLAB pentru implementarea algoritmului DVS de reducere la forma diagonală a unei matrice prin transformări ortogonale bilaterale. Pentru aceasta se va edita şi testa un program distinct pentru implementarea unei iteraţii DVS Golub-Kahan. Programul va afişa în faza iterativă vectorul g curent pentru vizualizarea procesului de anulare asimptotică a elementelor sale. Se va compara rezultatul obţinut cu cel oferit de funcţia MATLAB `svd`.

3. Folosind programul propriu de calcul al DVS, se va edita şi testa un program MATLAB pentru calculul rangului numeric (cu o toleranţă dată) al unei matrice.

4. Se va scrie un program MATLAB de calcul al soluţiei CMMP a unui sistem liniar în cazul cel mai general.

6.3.2 B. Acasă

1 Se va completa programul MATLAB de calcul al DVS, elaborat în laborator, cu ordonarea valorilor singulare în ordine descrescătoare. A nu se uita permutarea corespunzătoare a liniilor şi/sau a coloanelor matricelor de transformare.

2 Se va edita şi testa un program MATLAB pentru implementarea algoritmului de calcul al pseudoinversei unei matrice în cazul general. Se va verifica identitatea acestei pseudoinverse cu pseudinversele matricelor monice (epice) calculate pe baza triangularizărilor ortogonale la stânga (la dreapta) (sau a factorizărilor QR, respectiv LQ).

3 Comparaţi bazele ortogonale ale subspaţiilor $\mathrm{Im}A$, $\mathrm{Ker}A^T$, $\mathrm{Im}A^T$, $\mathrm{Ker}A$ calculate cu factorizările QR ale matricelor A şi A^T şi cu algoritmul DVS. Sunt aceleaşi sau diferă?

4 Fie două subspaţii liniare $\mathcal{S} = \mathrm{Im}S$ şi $\mathcal{T} = \mathrm{Im}T$, cu $S \in \mathbf{R}^{m\times n_S}$, $T \in \mathbf{R}^{m\times n_T}$, ale spaţiului $\mathbf{R}^m$. Se cer programe MATLAB pentru calculul unor baze ortogonale pentru subspaţiile intersecţie $\mathcal{S}\cap\mathcal{T} = \{x \in \mathbf{R}^m | x \in \mathcal{S}, x \in \mathcal{T}\}$, şi sumă $\mathcal{S}+\mathcal{T} = \{x \in \mathbf{R}^m | x = s+t, s \in \mathcal{S}, t \in \mathcal{T}\}$.

Bibliografie

B. Dumitrescu, C. Popeea, B. Jora, Metode de calcul numeric matriceal. Algoritmi fundamentali (cap.5), Ed. ALL, Bucureşti, 1998.