Tema nr. 5

Se dau $\boldsymbol{n}$ dimensiunea matricii, $\boldsymbol{\varepsilon}$ - precizia calculelor, $\boldsymbol{k}_{\max}$ – numărul maxim de iterații, matricea pătratică și nesingulară $A \in \mathbb{R}^{n \times n}$.

1. Folosind toate metodele iterative descrise mai jos, să se aproximeze inversa matricii $\boldsymbol{A}$. Pentru alegerea matricii inițiale V_0 să se utilizeze formulele (5), (6).
2. Pentru fiecare metodă în parte, la finalul calculelor să se afișeze numărul de iterații efectuate, iar în caz de convergență să se afișeze norma:

$$\left\| A * A_{\text{apx}}^{-1} - I_n \right\|_1$$

3. Pentru matricea:

$$A = \begin{pmatrix} 1 & 2 & 0 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 2 \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Să se deducă inductiv forma generală a inversei matricii $\boldsymbol{A}$ (prin rulări successive ale programului cu diverse valori pentru dimensiunea $\boldsymbol{n}$ a matricii $\boldsymbol{A}$).

Bonus (15pt): Să se adapteze unul din algoritmii de aproximare a inversei unei matrici, pentru matrici nepătratice, $A \in \mathbb{R}^{m \times n}$.

Metode iterative de aproximare a inversei unei matrici

Pentru aproximarea inversei unei matrici nesingulare $\boldsymbol{A}$, se construiește un șir de matrici $\{V_k\,;\,k \geq 0\}$ care converge la A^{-1}.

Metoda Schultz de construcție a șirului $\{V_k\,;\,k \geq 0\}$ este următoarea:

$$V_{k+1} = V_k\left(2I_n - AV_k\right), \quad k = 0,1,2,3,... \tag{1}$$

Această metodă mai poartă numele de algoritmul Hotelling-Bodewig sau metoda iterativă a hiper-puterii.

Li și Li au propus următoarele două metode iterative de aproximare a inversei:

$$V_{k+1} = V_k\left(3I_n - AV_k\left(3I_n - AV_k\right)\right), \quad k = 0,1,2,3,... \tag{2}$$

$$V_{k+1} = \left(I_n + \frac{1}{4}\left(I_n - V_k A\right)\left(3I_n - V_k A\right)^2\right)V_k, \quad k = 0,1,2,3,... \tag{3}$$

Pentru a asigura convergența șirului $\{V_n ;\ n\geq 0\}$ prima matrice din șir trebuie aleasă astfel încât să fie îndeplinită relația:

$$\left\| AV_0 - I_n \right\| < 1 \tag{4}$$

Modalități de alegere a lui V_0 care asigură convergența șirului:

1.

$$V_0 = \frac{A^T}{\left\|A\right\|_1 \left\|A\right\|_\infty} \tag{5}$$

$$\left\|A\right\|_1 = \max\left\{\sum_{i=1}^{n} |a_{ij}| \,;\ j = 1,2,...,n\right\},\ \left\|A\right\|_\infty = \max\left\{\sum_{j=1}^{n} |a_{ij}| \,;\ i = 1,2,...,n\right\} \tag{6}$$

2. Dacă matricea A are diagonala dominantă în raport cu liniile sau în raport cu coloanele:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \ ,\ \forall i = 1,2,...,n \quad \text{- dominanța în raport cu liniile}$$

$$|a_{jj}| > \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}| \ ,\ \forall j = 1,2,...,n \quad \text{- dominanța în raport cu coloanele}$$

alegerea care asigură convergența este:

$$V_0 = \mathbf{diag}\left(\frac{1}{a_{11}}, \frac{1}{a_{22}}, ..., \frac{1}{a_{nn}}\right).$$

3. Dacă matricea A este simetrică și pozitiv definită :

$$V_0 = \frac{1}{\|A\|_F} I_n \ , \quad \|A\|_F = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}} .$$

4.

$$V_0 = \alpha A^T \ , \ 0 < \alpha < \frac{2}{\|A\|_2^2} \ , \ \|A\|_2 = \sqrt{\rho(A^T A)} \ .$$

5.

$$V_0 = \alpha I_n \ , \ \alpha \in \mathbb{R} \ , \ \max\{|1 - \lambda_i \alpha|; \lambda_i \text{ valoare proprie pentru } A\} < 1 .$$

Algoritmul se oprește dacă :

1) $\| V_k - V_{k-1} \| < \varepsilon$ (sau $\|I_n - V_k A\| < \varepsilon$), caz în care matricea V_k aproximează inversa ($V_k \approx A^{-1}$);

   (se poate folosi oricare din normele matriciale descrise în (6))

2) $k > k_{\max}$ - s-a depășit un număr maxim de iterații stabilit inițial (nu s-a reușit aproximarea inversei);

3) $\| V_k - V_{k-1} \| > 10^{10}$ - șirul construit nu converge (nu s-a reușit aproximarea inversei).

Nu este nevoie de memorat tot șirul de matrici $\{V_k\}$, sunt suficiente doar două matrici, una pentru a memora V_k și una pentru V_{k+1}.

Pentru a calcula economic matricile de tipul $C = aI_n - AV$, se calculeză o singură dată în program matricea $B=(-A)$, apoi se calculează matricea produs $C=B*V$. Pentru a calcula matricea finală $C = aI_n - AV$, se adaugă a la toate elementele de pe diagonala matricii C ($c_{ii} = c_{ii} + a, \ i = 1,2,...,n$).

Schemă de implementare a unei metode iterative

$$V0 = V1 = \frac{A^T}{\|A\|_1 \|A\|_\infty} \quad ;$$

k=0;

$k_{max} = 10000$; // de exemplu

do

{

V0=V1;

calculează noul ***V1*** folosind ***V0*** cu una din formulele (1), (2) sau (3);

calculează $\Delta V = \| V1-V0 \|$;

k=k+1;

}

while $(\Delta V \geq \varepsilon$ **şi** $k \leq k_{max}$ **şi** $\Delta V \leq 10^{10})$

if $(\Delta V < \varepsilon)$ $V1 \approx A^{-1}$; // ***V1*** este aproximarea căutată a soluţiei

else *‚divergenţă’;*