Tema nr. 2

Se dau $\boldsymbol{n}$ dimensiunea sistemului, $\boldsymbol{\varepsilon}$ - precizia calculelor, matricea pătratică $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ şi un vector $\boldsymbol{s} \in \mathbb{R}^{n}$.

1. Să se calculeze vectorul $\boldsymbol{b} \in \mathbb{R}^{n}$ astfel:

$$b_i = \sum_{j=1}^{n} s_j a_{ij} \quad , \; i = 1,\ldots,n$$

2. Să se implementeze descompunerea $\boldsymbol{QR}$ a matricii $\boldsymbol{A}$ folosind algoritmul lui Householder.
3. Să se rezolve sistemul liniar:

$$Ax = b \quad ,$$

   folosind descompunerea $\boldsymbol{QR}$ din una din bibliotecile menţionate în pagina laboratorului (se obţine soluţia $\boldsymbol{x_{QR}}$) şi descompunerea $\boldsymbol{QR}$ calculată la punctul 2. (se obţine soluţia $\boldsymbol{x_{Householder}}$). Pentru $\boldsymbol{n \geq 250}$, să se compare cele două rezolvări din punct de vedere al timpului de execuţie (în acest caz matricile se vor genera aleator).
4. Să se calculeze şi să se afişeze următoarele erori ($\|\ \|_2$ este norma euclidiană):

$$\| A^{init} x_{Householder} - b^{init} \|_2 \quad ,$$

$$\| A^{init} x_{QR} - b^{init} \|_2 \quad ,$$

$$\frac{\| x_{Householder} - s \|_2}{\| s \|_2} \quad ,$$

$$\frac{\| x_{QR} - s \|_2}{\| s \|_2} \, .$$

Vectorul $\boldsymbol{y=Ax}$ se calculează folosind formula :

$$x \in \mathbb{R}^n \, , A \in \mathbb{R}^{n \times n} \, , \; y = Ax \in \mathbb{R}^n , \; y = \left(y_i\right)_{i=1,n} , \; y_i = \sum_{j=1}^{n} a_{ij} x_j \, , \; \forall i = 1,\ldots,n \, .$$

Norma Euclidiană a unui vector z se calculează folosind relația:

$$z \in \mathbb{R}^n \; , \;\; z = \left(z_i\right)_{i=1,n} \;\; , \;\; |z| = \|z\|_2 = \sqrt{\sum_{i=1}^{n} | z_i |^2}$$

Observație: Precizia calculelor ***ε***, este un număr pozitiv de forma:

$$\varepsilon = 10^{-m} \text{ cu } m = 5, 6, ..., 10, ... \text{ la alegere}$$

valoare care este dată de intrare în program (se citește de la tastatură sau din fișier) la fel ca și dimensiunea ***n*** a datelor. Acest număr se folosește atunci când testăm dacă o variabilă are valoarea 0 sau nu înaintea unei operații de împărțire.

Dacă vrem să efectuăm operația de împărțire $s = \frac{1}{v}$ unde $v \in \mathbb{R}$ nu vom scrie:

```
if (v != 0)
{
      s=1/v;
}
```

ci vom scrie în program:

```
if ( Math.Abs(v) > eps )
{
      s=1/v;
}
```

Metoda substituției inverse

Fie sistemul liniar:

$$Ax = b \tag{1}$$

unde matricea sistemului A este superior triunghiulară. Pentru a găsi soluția unică a sistemului (1), trebuie ca matricea să fie nesingulară. Determinantul matricilor triunghiulare este dat de formula:

$$\det A = a_{11} \cdot a_{22} \cdots a_{nn}$$

Prin urmare, pentru rezolvarea sistemului (1) vom presupune că:

$$\det A \neq 0 \iff a_{ii} \neq 0 \ \forall i = 1, ..., n$$

Vom considera, în continuare sistemul (1) cu matrice superior triunghiulară :

$$\begin{array}{l} a_{11}x_1 + \cdots + a_{1i}x_i + \cdots + a_{1n-1}x_{n-1} + a_{1n}x_n = b_1 \\ \ddots \\ \qquad a_{ii}x_i + \cdots + a_{in-1}x_{n-1} + a_{in}x_n = b_i \\ \qquad \ddots \\ \qquad\qquad a_{n-1n-1}x_{n-1} + a_{n-1n}x_n = b_{n-1} \\ \qquad\qquad\qquad a_{nn}x_n = b_n \end{array}$$

Necunoscutele $x_1, x_2, \ldots, x_n$ se deduc pe rând, folosind ecuaţiile sistemului, de la ultima către prima.

Din ultima ecuaţie găsim x_n:

$$x_n = \frac{b_n}{a_{nn}}$$

Folosind valoarea lui x_n dedusă mai sus, din penultima ecuaţie obţinem:

$$x_{n-1} = \frac{b_{n-1} - a_{n-1n}x_n}{a_{n-1n-1}}$$

Când ajungem la ecuaţia i:

$$a_{ii}x_i + a_{ii+1}x_{i+1} + \cdots + a_{in}x_n = b_i$$

se cunosc deja $x_{i+1}, x_{i+2}, \ldots, x_n$ de unde ducem:

$$x_i = \frac{b_i - a_{ii+1}x_{i+1} - \cdots - a_{in}x_n}{a_{ii}}$$

Din prima ecuaţie găsim valoarea lui x_1:

$$x_1 = \frac{b_1 - a_{12}x_2 - \cdots - a_{1n}x_n}{a_{11}}$$

Procedeul descris mai sus se numeşte ***metoda substituţiei inverse*** pentru rezolvarea sistemelor liniare cu matrice superior triunghiulară:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}, i = n, n-1, \cdots, 2, 1.$$

Rezolvarea sistemelor liniare folosind descompunerea *QR*

Fie A o matrice reală pătratică de dimensiune n. Presupunem că pentru matricea A avem o descompunere de forma:

$$A=Q*R$$

unde Q este matrice ortogonală ($Q^TQ = QQ^T = I_n$) iar R este matrice superior triunghiulară. Având o asemenea descompunere pentru matricea A, rezolvarea sistemului $Ax=b$ se reduce la rezolvarea sistemului superior triunghiular $Rx=Q^Tb$.

$$Ax=b \quad \leftrightarrow \quad QRx = b \quad \leftrightarrow \quad Q^TQRx = Q^T b \quad \leftrightarrow \quad Rx=Q^T b$$

$$Ax=b \qquad \Leftrightarrow \qquad Rx=Q^T b$$

Algoritmul lui Householder

Petru a aduce sistemul $Ax=b$ la forma $Rx = Q^Tb$ se folosesc matricile de reflexie. O matrice de reflexie $P = (p_{ij})_{i,j=1,n}$ are următoarea formă:

$$P = I_n - 2vv^T \quad , \quad v \in \mathbb{R}^n, \ \|v\|_2 = |\, v \,| = \sqrt{\sum_{i=1}^{n} v_i^2} = 1 \, .$$

(unde cu I_n am notat matricea unitate de dimensiune n)

Se poate arăta că matricile de reflexie sunt simetrice şi ortogonale :

$$P = P^T \quad , \quad P^2 = I_n \, .$$

Algoritmul Householder de calcul al descompunerii QR se desfăşoară în $(n-1)$ paşi. La pasul r se transformă coloana r a matricii A în formă superior triunghiulară fără a modifica primele $(r-1)$ coloane. La acest pas se obţine coloana r a matricii R. Calculul matricii R se poate face direct în matricea A (de fapt se transformă matricea A într-o matrice superior triunghiulară). În acelaşi timp, se fac transformările necesare asupra vectorului termenilor liberi şi se pot efectua transformări pentru a calcula matricea Q^T.

Pasul ***r (r=1,2,…,n-1)***

La acest pas, matricea A are primele ***(r-1)*** coloane în formă superior triunghiulară şi se foloseşte o matrice de reflexie pentru a transforma coloana r a matricii A în formă superior triunghiulară (fără a le modifica pe primele ***(r-1)***).
Pasul r constă în următoarele calcule :

$$A = P_r A$$
$$b = P_r b$$

unde matricea P_r se construieşte astfel :

$$P_r = I_n - \frac{1}{\beta} u u^T$$

$$\beta = \sigma - k a_{rr}, \ \ \sigma = \sum_{j=r}^{n} a_{jr}^2 \ , \ \ k = -\mathrm{semn}(a_{rr})\sqrt{\sigma}$$

$$u = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix}, \qquad \begin{array}{c} u_i = 0, \ i = 1,\ldots,r-1, \\ u_r = a_{rr} - k, \\ u_i = a_{ir}, i = r+1,\ldots,n \end{array}$$

Matricea $V = uu^T \in \mathbb{R}^{n\times n}$, $V = \left(v_{ij}\right)_{i,j=1,\ldots,n}$ are următoarele elemente:

$$v_{ij} = \begin{cases} 0 & \text{pentru } i = 1,\ldots,r-1 \ , \ \ j = 1,\ldots,n \\ 0 & \text{pentru } i = r,\ldots,n \ , \ \ j = 1,\ldots,r-1 \\ u_i u_j & \text{pentru } i = r,\ldots,n \ , \ \ j = r,\ldots,n \end{cases}$$

Prin urmare, matricea de reflexie $\boldsymbol{P_r} = (\boldsymbol{p_{ij}})_{i,j=1,n}$ are următoarele elemente:

$$p_{ij} = \begin{cases} 0 & \text{pentru } i = 1,\dots,r-1\,,\; j = 1,\dots,n,\; i \neq j \\ 1 & \text{pentru } i = 1,\dots,r-1,\; j = i \\ 0 & \text{pentru } i = r,\dots,n\,,\; j = 1,\dots,r-1 \\ -f\,u_i u_j & \text{pentru } i = r,\dots,n\,,\; j = r,\dots,n,\; i \neq j \\ 1 - f\,u_i^2 & \text{pentru } i = r,\dots,n\,,\; j = i \end{cases} \quad , \quad f = \frac{1.0}{\beta}$$

La sfârşitul algoritmului de mai jos, matricea superior triunghiulară ***R*** se găseşte în matricea ***A***, iar transpusa matricii ***Q*** în matricea $\overline{q}$. În vectorul ***b*** vom avea $\boldsymbol{Q^T b^{init}}$. Verificarea faptului că matricea iniţială nu este singulară, se face testând dacă toate elementele de pe diagonala matricii ***R*** (sau ***A***) sunt nenule.

Algoritmul Householder pentru factorizarea ***A=QR***

$\widetilde{Q} = I_n$;

for $r = 1,\dots,n-1$

// **construcţia matricii** P_r – **constanta** β **şi vectorul** u

- $\sigma = \sum_{i=r}^{n} a_{ir}^2$;
- **if** ($\sigma \le \varepsilon$) **break** ; // $r = r+1 \leftrightarrow P_r = I_n$ (A **singulară**)
- $k = \sqrt{\sigma}$;
- **if** ($a_{rr} > 0$) $k = -k$;
- $\beta = \sigma - k\, a_{rr}$;
- $u_r = a_{rr} - k$; $u_i = a_{ir}$, $i = r+1,\dots,n$;

// $A = P_r * A$

// **transformarea coloanelor** $j = r+1,\dots,n$

- **for** $j = r+1,\dots,n$

  $* \gamma = (\gamma_j / \beta) = (Ae_j, u) / \beta = (\sum_{i=r}^{n} u_i a_{ij}) / \beta$;

  $*$ **for** $i = r,\dots,n$

  $a_{ij} = a_{ij} - \gamma * u_i$;

// **transformarea coloanei** r **a matricii** A

- $a_{rr} = k$; $a_{ir} = 0$, $i = r+1,\dots,n$;

// $b = P_r * b$

- $\gamma = (\gamma / \beta) = (b, u) / \beta = (\sum_{i=r}^{n} u_i b_i) / \beta$;
- **for** $i = r,\dots,n$ $\quad b_i = b_i - \gamma * u_i$;

// $\widetilde{Q} = P_r * \widetilde{Q}$

- **for** $j = 1,\dots,n$

  $* \gamma = (\widetilde{Q} e_j, u) / \beta = (\sum_{i=r}^{n} u_i \tilde{q}_{ij}) / \beta$;

  $*$ **for** $i = r,\dots,n$

  $\tilde{q}_{ij} = \tilde{q}_{ij} - \gamma * u_i$;