Tema nr. 4

În fişierele matrar1.txt, matrar2.txt, matrar3.txt postate pe pagina laboratorului, sunt memorate pentru 3 sisteme liniare cu matrice rară (cu *'puţine'* elemente $a_{ij} \neq 0$), $Ax = b$, următoarele elemente (în această ordine):

- n dimensiunea sistemului,
- b_i , $i=1,2, \dots , n$ elementele vectorului termenilor liberi $b \in \mathbb{R}^n$
- $a_{ij} \neq 0$, i, j - elementele nenule din matricea rară $A \in \mathbb{R}^{n \times n}$, indicii de linie şi de coloană ai respectivului element. Într-un acelaşi fişier, pot exista mai multe intrări (linii în fişier, nu neapărat consecutive) cu valori nenule diferite şi acelaşi indice de linie şi acelaşi indice de coloană:

  val_1 , i, j
  val_2 , i, j
  ...
  val_p , i, j

  cu $val_k \neq val_r$. O astfel de situaţie are următoarea semnificaţie :

$$a_{ij} = \sum_{k=1}^{p} val_k$$

1. Folosind fişierele ataşate, să se citească dimensiunea sistemului, vectorul termenilor liberi şi să se genereze, folosind schema economică de memorare, vectorii necesari pentru memorarea matricii rare (schema economică de memorare este descrisă mai jos). Se presupune că elementele nenule ale matricii sunt plasate aleator în fişier (nu sunt ordonate după indicii de linie sau de coloană, sau altfel). Precizia calculelor este $\varepsilon = 10^{-p}$.

2. Cu această memorare rară a matricii $\boldsymbol{A}$ să se aproximeze soluţia sistemului liniar:

$$\boldsymbol{Ax=b} \qquad \textbf{(1)}$$

   folosind metoda iterativă SOR. Parametrul $\boldsymbol{\omega}$ care apare în descrierea metodei SOR este dată de intrare, $\boldsymbol{\omega \in (0,2)}$.

3. Să se verifice soluţia calculată afişând norma:

$$\left\| \boldsymbol{A x_{SOR} - b} \right\|_1$$

   unde $\boldsymbol{x_{SOR}}$ este aproximarea soluţiei exacte obţinută cu algoritmul SOR.

4. În toate calculele care includ matricea $\boldsymbol{A}$, se cere să se utilizeze memorarea rară a matricii (să nu se aloce în program nici o matrice clasică).

Metode iterative de rezolvare a sistemelor liniare

Pp. că **det** $\boldsymbol{A \neq 0}$, vom nota soluţia exactă a sistemului (1) cu $\boldsymbol{x^*}$:

$$\boldsymbol{x^* := A^{-1} b.}$$

Metodele iterative de rezolvare a sistemelor liniare au fost deduse pentru sistemele de dimensiune '*mare*' ($\boldsymbol{n}$ '*mare*'), cu matricea sistemului $\boldsymbol{A}$, matrice rară (cu *'puţine'* elemente $\boldsymbol{a_{ij}}$ nenule). În cazul metodelor iterative matricea $\boldsymbol{A}$ nu se transformă (ca în cazul agoritmului de eliminare Gauss sau a descmpunerilor $\boldsymbol{LU}$ sau a factorizărilor $\boldsymbol{QR}$) ci sunt folosite doar elementele nenule ale matricii pentru aproximarea soluţiei exacte $\boldsymbol{x^*}$. Pentru matricile rare se folosesc scheme de memorare economice specifice.

Pentru a aproxima soluţia $\boldsymbol{x}^*$ se construieşte un şir de vectori $\left\{\boldsymbol{x}^{(k)}\right\} \subset \mathbb{R}^n$ care, în anumite condiţii, converge la soluţia exactă $\boldsymbol{x}^*$ a sistemului (1):

$$\boldsymbol{x}^{(k)} \rightarrow \boldsymbol{x}^* \ , \ \textbf{pentru } \boldsymbol{k} \rightarrow \infty$$

Metode de construcţie a şirului $x^{(k)}$

Se consideră şirul:

$$x^{(k+1)}=Mx^{(k)} + d \, , \quad \textbf{unde } M \in R^{n\times n} \ \textbf{şi} \ d \in R^n \ , x^{(0)} \ \textbf{ales arbitrar.}$$

Matricea $\boldsymbol{M}$ şi vectorul $\boldsymbol{d}$ depind de datele de intrare (matricea $\boldsymbol{A}$ şi vectorul $\boldsymbol{b}$) se aleg astfel încât şirul $\{x^{(k)}\}$ să conveargă la soluţia sistemului liniar considerat.

Vectorul $\boldsymbol{x}^{(0)}$ se iniţializează, de obicei, cu 0:

$$\boldsymbol{x}_i^{(0)} = \boldsymbol{0} \ , \ \boldsymbol{i} = \boldsymbol{1},\ldots,\boldsymbol{n}$$

Şirul $\{x^{(k)}\}$ astfel construit este convergent dacă şi numai dacă raza spectrală a matricii $\boldsymbol{M}$ este subunitară. Se obţin diverse metode iterative de aproximare a soluţiei sistemului liniar (1) pentru diversele alegeri ale matricii $\boldsymbol{M}$ şi a vectorului $\boldsymbol{d}$.

Metode SOR

Fie $\omega \in (0,2)$ o constntă dată. Vom presupune că toate elementele diagonale ale matricii A sunt nenule:

$$a_{ii} \neq 0\ ,\ \ i=1,\ldots,n$$

Când se citeşte matricea din fişier, se cere să se verifice dacă elementele diagonale ale matricii sunt nenule ($|a_{ii}| > \varepsilon,\ \forall\, i$). Dacă există un element diagonal nul, nu se poate rezolva sistemul liniar folosind această metoda iterativă.
Şirul de vectori generat de metoda SOR este următorul:

$$x^{(0)} \in \mathbb{R}^n \text{ dat } (\text{de obicei } x^{(0)} = 0)\ ,\ k = 0,1,\ldots$$

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)\ ,\ i = 1,2,\ldots,n. \tag{2}$$

Formula de calcul de mai sus trebuie adaptată noului tip memorare a matricii A. În sumele de mai sus sunt necesare doar elemntele a_{ij} nenule. Pentru un calcul rapid al componentei i a vectorului de aproximare $x_i^{(k+1)}$ avem nevoie să accesăm uşor elementele liniei i a matricii A, din acest motiv în schema economică de memorare vom ţine cont de acest lucru.
Pentru metodele iterative de rezolvare a sistemelor liniare convergenţa sau divergenţa şirului $\{x^{(k)}\}$ nu depinde de alegerea iteraţiei iniţiale $x^{(0)}$.

Pentru a aproxima soluţia x^* trebuie să calculăm un termen al şirului $x^{(k)}$ pentru k suficient de mare. Se ştie că, dacă diferenţa dintre doi termeni consecutivi ai şirului $\{x^{(k)}\}$ devine suficient de *'mică'*, atunci ultimul vector calculat este *'aproape'* de soluţia căutată:

$$\| \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} \| \le \boldsymbol{\varepsilon} \Rightarrow \| \boldsymbol{x}^{(k)} - \boldsymbol{x}^{*} \| \le \boldsymbol{c\varepsilon}\ ,\ \boldsymbol{c} \in \mathbb{R}_{+} \rightarrow \ \boldsymbol{x}^{(k)} \approx \boldsymbol{x}^{*} \quad (3)$$

Nu este nevoie să memorăm toţi vectorii calculaţi ai şirului $\{x^{(k)}\}$ ci avem nevoie doar de ultimul vector, cel care satisface prima inegalitate din relaţia (3) ($\| \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} \| \le \boldsymbol{\varepsilon}$). În program s-ar putea utiliza doar doi vectori:

$\boldsymbol{x}^{c}$ pentru vectorul $\boldsymbol{x}^{(k+1)}$ şi $\boldsymbol{x}^{p}$ pentru vectorul $\boldsymbol{x}^{(k)}$.

Schemă de implementare a unei metode iterative

$x^c = x^p = 0$;

$k=0$;

do

{

$x^p = x^c$;

calculează noul x^c folosind x^p (cu formula (2));

calculează $\Delta x = \| x^c - x^p \|$;

$k=k+1$;

}

while ($\Delta x \ge \varepsilon$ şi $k \le k_{max}$ şi $\Delta x \le 10^8$)

if ($\Delta x < \varepsilon$) $x^c \approx x^*$; // x^c este aproximarea căutată a soluţiei

else *‚divergenţă’;*

Memorarea matricilor rare (schema de memorare economică)

Pentru matricile rare se memorează doar elementele nenule ale matricii şi informaţii privind indicii de linie şi de coloană ale respectivelor elemente astfel încât să putem reface toată informaţia din matricea în formă clasică.

Vom nota cu ***NN*** numărul de elemente nenule ale matricii ***A***. Matricea ***A*** se memorează folosind 2 vectori:

- un vector pentru elementele nenule ale matricii ***A, valori*** – vector de elemente reale de dimensiune ***NN+1***,
- un vector care conţine informaţii despre indici, ***indici*** - vector de numere naturale (întregi) de dimensiune ***NN+1***.

Schema de memorare propusă trebuie să ţină cont de faptul că avem nevoie de acces rapid la liniile matricii ***A***. Vom memora elementele nenule ale matricii ***A*** în ordinea crescătoare a indicilor de linie, în cadrul liniei nu e neapărat ca elementele să fie memorate în ordinea crescătoare a indicilor de coloană.

Vectorul ***valori*** conţine pe primele ***n*** poziţii elementele diagonale ale matricii ***A*** iar de la poziţia ***n+2*** sunt memorate celelalte elemente nenule ale matricii ***A*** în ordinea indicilor de linie (poziţia ***n+1*** este nefolosită).

$$\boldsymbol{valori}_i = \begin{cases} \boldsymbol{a}_{ii} & \text{pentru } \boldsymbol{i = 1, 2, \ldots, n} \\ * & \boldsymbol{i = n+1} \text{ (nefolosit)} \\ \boldsymbol{a}_{pj} \neq \boldsymbol{0} & \boldsymbol{i \geq n+2} \text{ (in ordinea crescatoare a liniilor, a indicelui } \boldsymbol{p}\text{)} \end{cases}$$

Vectorul ***indici*** conţine pe primele ***n*** poziţii, indicii din vectorul ***valori*** unde încep liniile matricii ***A***. Pe poziţia ***(n+1)*** se reţine valoarea ***NN+2*** (poziţia unde ar începe linia ***(n+1)***, dacă ar exista o asemenea linie). De la poziţia ***n+2*** sunt memoraţi indicii de coloană ai elementelor corespunzătoare din vectorul ***valori***.

$$
\boldsymbol{indici}_i = \begin{cases} \boldsymbol{l_i} & \text{pozitia in vectorul } \boldsymbol{valori} \text{ unde incepe linia } \boldsymbol{i},\ \boldsymbol{i = 1,2,...,n} \\ \boldsymbol{NN+2} & \boldsymbol{i = n+1} \text{ (nefolosit)} \\ \boldsymbol{j} & \boldsymbol{i \geq n+2} \text{ (indicele de coloana al elementului din } \boldsymbol{valori_i}\text{)} \end{cases}
$$

Avem următoarea proprietate care ne ajută să accesăm rapid elementele liniilor matricii A :
Vectorul ***indici*** are următoarea proprietate pentru ***i=1,2,...,n***:

$\boldsymbol{indici_{i+1} - indici_i}$ = numărul de elemente nenule de pe linia ***i*** (cu excepţia celui diagonal), ***i=1,2,...,n***

Exemplu:

Matricea:

$$
A = \begin{pmatrix} \mathbf{102.5} & \mathbf{0.0} & \mathbf{2.5} & \mathbf{0.0} & \mathbf{0.0} \\ \mathbf{3.5} & \mathbf{104.88} & \mathbf{1.05} & \mathbf{0.0} & \mathbf{0.33} \\ \mathbf{0.0} & \mathbf{0.0} & \mathbf{100.0} & \mathbf{0.0} & \mathbf{0.0} \\ \mathbf{0.0} & \mathbf{1.3} & \mathbf{0.0} & \mathbf{101.3} & \mathbf{0.0} \\ \mathbf{0.73} & \mathbf{0.0} & \mathbf{0.0} & \mathbf{1.5} & \mathbf{102.23} \end{pmatrix}
$$

se memorează astfel:

n=5 , NN=12

$$
\boldsymbol{valori} = (\ \mathbf{102.5 \quad 104.88 \quad 100.0 \quad 101.3 \quad 102.23 \quad *} \\ \mathbf{2.5 \quad 1.05 \quad 3.5 \quad 0.33 \quad 1.3 \quad 0.73 \quad 1.5}\ )
$$

$$
\boldsymbol{indici} = \left(\mathbf{7 \quad 8 \quad 11 \quad 11 \quad 12 \quad 14 \quad 3 \quad 3 \quad 1 \quad 5 \quad 2 \quad 1 \quad 4}\right)
$$

Presupunem că:

$$\boldsymbol{x}^{(0)} = \begin{pmatrix} \mathbf{1.0} \\ \mathbf{2.0} \\ \mathbf{3.0} \\ \mathbf{4.0} \\ \mathbf{5.0} \end{pmatrix} \quad , \quad \boldsymbol{b} = \begin{pmatrix} \mathbf{6.0} \\ \mathbf{7.0} \\ \mathbf{8.0} \\ \mathbf{9.0} \\ \mathbf{1.0} \end{pmatrix}$$

Pentru metoda SOR cu ***ω=1, v***om calcula mai jos componentele $x_1^{(1)}$, $x_2^{(1)}$ şi $x_3^{(1)}$ cu memorarea clasică (cu matrici $n \times n$) şi cu schema de memorare rară propusă.

$x_1^{(1)}$ **(**varianta clasică)

$$= (b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - a_{14}x_4^{(0)} - a_{15}x_5^{(0)}) \,/\, a_{11} =$$

$$= (6.0 - 0.0 * 2.0 - 2.5 * 3.0 - 0.0 * 4.0 - 0.0 * 5.0) \,/\, 102.5$$

(varianta economică foloseşte elementele de pe linia 1)

$$= (6.0 - 2.5 * 3.0) \,/\, 102.5 = -\,0.01463414...$$

$x_2^{(1)}$ **(**varianta clasică)

$$= (b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)} - a_{24}x_4^{(0)} - a_{25}x_5^{(0)}) \,/\, a_{22} =$$

$$= (7.0 - 3.5 * (-0.01463414...) - 1.05 * 3.0 - 0.0 * 4.0 - 0.33 * 5.0) \,/\, 104.88$$

(varianta economică foloseşte elementele nenule de pe linia 2)

$$= (7.0 - 1.05 * 3.0 - 3.5 * (-0.01463414...) - 0.33 * 5.0) \,/\, 104.88$$

$x_3^{(1)}$ (varianta clasică)

$$= (b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)} - a_{34}x_4^{(0)} - a_{35}x_5^{(0)}) / a_{33} =$$

$$= (8.0 - 0.0 * (-0.01463414...) - 0.0 * x_2^{(1)} - 0.0 * 4.0 - 0.0 * 5.0) / 100.0$$

(varianta economică foloseşte elementele nenule de pe linia 3)

$$= (8.0) / 100.00$$

$x^{(k+1)}[i]$ (varianta economică foloseşte elementele nenule de pe linia i)=

$$= \frac{\left(b[i] - \sum (\text{valoare nediag} \neq \mathbf{0} \text{ de pe linia } i) * x^{(?)}[\text{indice de col. corespunzător valorii}]\right)}{\text{valoarea elementului diagonal de pe lina } i \ (valori_i)}$$

Sistemul memorat în fisierul matrar1.txt are soluţia $x_i = i$, $\forall i$, cel memorat în fişierul matrar2.txt are soluţia $x_i = \frac{1}{3}$, $\forall i$ iar soluţia sistemului din matrar3.txt este $x_i = 1$, $\forall i$.