LABORATOR#1

Fie $\mathbf{A} \in \mathscr{M}_n(\mathbb{R})$ inversabilă, $\mathbf{b} \in \mathbb{R}^n$ şi sistemul de ecuaţii liniare asociat

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}\,. \tag{1}$$

EX#1 Să se construiască funcţia `Jacobi` care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda Jacobi*, are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $= 0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $= 1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;
- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#2 Să se construiască funcţia `GS` (`GSb`) care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda Gauss-Seidel* (*metoda Gauss-Seidel descendentă*), are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $= 0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $= 1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;
- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#3 Să se construiască funcţia `JOR` care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda Jacobi suprarelaxată*, are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- ω – parametrul de relaxare;
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $=0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $=1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;
- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#4 Să se construiască funcţia `SOR` (`SORb`) care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda de suprarelaxare succesivă* (*metoda de suprarelaxare succesivă descendentă*), are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- ω – parametrul de relaxare;
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $=0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $=1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;

- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#5 Să se construiască funcţia `SGS` care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda Gauss-Seidel simetrică*, are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $= 0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $= 1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;
- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#6 Să se construiască funcţia `SSOR` care rezolvă numeric sistemul de ecuaţii liniare (1) prin *metoda de suprarelaxare succesivă simetrică*, are ca date de intrare:

- $\mathbf{A}$ – matricea sistemului (1);
- $\mathbf{b}$ – membrul drept al sistemului (1);
- $\mathbf{x}^{(0)}$ – aproximarea iniţială a soluţiei sistemului (1);
- ω – parametrul de relaxare;
- `ITMAX` – numărul maxim de iteraţii admise;
- `TOL` – toleranţa/precizia soluţiei numerice;
- `OPT` – parametru opţional care face selecţia erorii folosite, e.g. `OPT` $= 0$ pentru incrementul/corecţia aproximării de la pasul k, $\delta\mathbf{x}^{(k-1)} := \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, respectiv `OPT` $= 1$ pentru eroarea reziduală de la pasul k, $\mathbf{r}^{(k)} := \mathbf{b} - \mathbf{A}\,\mathbf{x}^{(k)}$;

şi care returnează:

- $\mathbf{x}^{(\mathrm{num})}$ – aproximarea soluţiei sistemului (1), obţinută folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa `TOL`;
- k_f – numărul de iteraţii necesare pentru obţinerea aproximąrii soluţiei sistemului (1), folosind criteriul corespunzător parametrului `OPT` şi cu toleranţa dorită.

EX#7 Testaţi funcţiile `Jacobi`, `GS`, `GSb`, `JOR` cu $\omega \in (0, 1]$, respectiv `SOR`, `SORb`, `SGS` şi `SSOR` cu $\omega \in (0, 2)$, $\texttt{ITMAX} = 1.000$, $\texttt{TOL} = 10^{-14}$ şi $\texttt{OPT} = 0$, respectiv $\texttt{OPT} = 1$, pentru următoarele matrice şi vectori:

(a) **Test#1:** matricea $\mathbf{A}$ nu este pătratică

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 3 \\ 2 & 2 & 3 \end{bmatrix} \qquad \text{şi} \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}.$$

(b) **Test#2:** matricea $\mathbf{A}$ şi vectorul $\mathbf{b}$ nu sunt compatibili

$$\mathbf{A} = \begin{bmatrix} 3 & 3 & -6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{bmatrix} \qquad \text{şi} \qquad \mathbf{b} = \begin{bmatrix} 0 \\ -5 \\ 3 \end{bmatrix}.$$

(c) **Test#3:** matricea $\mathbf{A}$ nu este inversabilă

$$\mathbf{A} = \begin{bmatrix} 4 & 1 & 1 \\ 2 & -9 & 0 \\ 0 & -6 & -8 \end{bmatrix} \qquad \text{şi} \qquad \mathbf{b} = \begin{bmatrix} 6 \\ -7 \\ -14 \end{bmatrix}.$$

(d) **Test#4:** sistemul (1) are soluţie unică $\mathbf{x}^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\mathsf{T}$ şi metodele iterative sunt divergente

$$\mathbf{A} = \begin{bmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & 1 & 2 \end{bmatrix} \qquad \text{şi} \qquad \mathbf{b} = \begin{bmatrix} 7 \\ 13 \\ 2 \end{bmatrix}.$$

(e) **Test#5:** sistemul (1) are soluţie unică $\mathbf{x}^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\mathsf{T}$ şi metodele iterative sunt convergente

$$\mathbf{A} = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 3 \end{bmatrix} \qquad \text{şi} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}.$$

La fiecare pas al iteraţiei, afişaţi pe aceeaşi linie următoarele:

- iteraţia actuală k, $k \geq 0$, formatată ca număr întreg;
- norma euclidiană a erorii absolute a aproximării actuale în raport cu cea anterioară, i.e. a incrementului/corecţiei de la iteraţia actuală, $\|\delta\mathbf{x}^{(k-1)}\|_2 := \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2$, $k \geq 1$, formatată ca număr scris în formă şiinţifică standard cu 4 zecimale;
- norma euclidiană a erorii reziduale $\|\mathbf{r}^{(k)}\|_2 := \|\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}\|_2$, $k \geq 0$, (formatată ca număr scris în formă şiinţifică standard cu 4 zecimale).

OBSERVAŢIE: Studenţii cu experienţă de programare cel puţin medie pot considera **EX#1** şi **EX#2** drept cazuri particulare pentru **EX#3** cu $\omega = 1$, respectiv **EX#4** cu $\omega = 1$.