METODE NUMERICE: Laborator #5
Metode iterative pentru rezolvarea sistemelor: Jacobi, Gauss-Siedel, Suprarelaxare

Titulari curs: *Florin Pop, George-Pantelimon Popescu*

Responsabil Laborator: **Mădălina-Andreea Grosu**

Obiective Laborator. Motivaţie

În urma laboratorului studentul va fi capabil să rezolve sisteme de ecuaţii liniare utilizând metode iterative, să observe diferenţele dintre metodele directe şi iterative.

Metodele exacte de rezolvare a sistemelor de ecuaţii liniare, având complexitate $O(n^3)$, au aplicabilitate limitată la ordine de sisteme ce nu depăşesc 1000. Pentru sisteme de dimensiuni mai mari se utilizează metode cu complexitate $O(n^2)$ într-un singur pas de iteraţie. Acestea utilizează relaţii de recurenţă, care prin aplicare repetată furnizează aproximaţii cu precizie controlată a soluţiei sistemului.

Noţiuni teoretice

La metodele iterative se încearcă transformarea sistemului $Ax = b$ la forma $x = Gx + c$. Pornindu-se cu o aproximaţie iniţială $x^{(0)}$ a soluţiei se generează, folosind o relaţie de recurenă de forma:

$$x^{(p+1)} = Gx^{(p)} + c \tag{1}$$

unde

- $x^{(0)}, x^{(1)}, ..., x^{(p)}, ...$ sunt aproximările soluţiei;
- G reprezintă matricea de iteraţie a sistemului;
- c reprezintă vectorul de iteraţie.

O metodă este convergentă dacă este stabilă şi consistentă. Condiţia necesară şi suficientă de convergenţă este:

$$\rho(G) < 1 \tag{2}$$

unde $\rho(G) = \max(|\lambda_1|, |\lambda_2|, ..., |\lambda_n|)$ reprezintă raza spectrală a matricei de iteraţie G şi $\lambda_i, i = 1 : n$ reprezintă valorile proprii ale matricei.

Metodele iterative se bazează pe descompunerea matricei A sub forma $A = N - P$. Atunci sistemul devine

$(N - P)x = b$

$x = N^{-1}Px + N^{-1}b$

de unde rezultă relaţia de recurenţă

$$x^{(p+1)} = N^{-1}Px^{(p)} + N^{-1}b \tag{3}$$

de unde putem identifica $G = N^{-1}P$ şi $c = N^{-1}b$.

Se partiţionează matricea A punând în evidenţă o matrice diagonală D, o matrice strict triunghiular inferioară L şi o matrice strict triunghiular superioară U:

$$A = D - L - U. \tag{4}$$

Metoda Jacobi

În metoda Jacobi se aleg:

$N = D$

$P = L + U$

$G_J = D^{-1}(L + U)$

Soluţia sistemului se poate scrie sub forma:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1}^{n} a_{ij}x_j^{(p)}}{a_{ii}}; i \neq j \tag{5}$$

Metoda Gauss-Seidel

La această metodă se alege:

$N = D - L$

$P = U$

$G_{GS} = (D - L)^{-1}U$

Soluţia sistemului se poate scrie sub forma:

$$x_i^{(p+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(p+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(p)}}{a_{ii}} \tag{6}$$

Metoda suprarelaxării

Pentru găsirea unei descompuneri cât mai rapid convergente se introduce parametrul de relaxare ω:

$$A = N - P = N - \omega N - P + \omega N = (1 - \omega)N - (P - \omega N) = N(\omega) - P(\omega)$$

de unde obţinem:

$N(\omega) = (1 - \omega)N$

$P(\omega) = P - \omega N$

$G(\omega) = N^{-1}(\omega)P(\omega) = \frac{N^{-1}}{1-\omega}(P - \omega N) = \frac{N^{-1} - \omega I_n}{1-\omega}$

Condiţia de stabilitate impune $\omega \in (0,2)$.

În practică se face o altă alegere, astfel:

$N(\omega) = \frac{1}{\omega}D - L, \quad P(\omega) = (\frac{1}{\omega} - 1)D + U, \quad G_\omega = (D - \omega L)^{-1}[(1-\omega)D + \omega U]$

Soluţia sistemului se poate scrie sub forma:

$$x_i^{(p+1)} = \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(p+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(p)}}{a_{ii}} + (1-\omega) x_i^{(p)} \tag{7}$$

Dacă se alege $\omega = 1 \Rightarrow$ metoda Gauss-Seidel.

Probleme rezolvate

1. Să se rezolve sistemul folosind metoda Gauss-Seidel:

$$\begin{cases} 4x_1 & + & x_2 & - & x_3 & = & 3 \\ 2x_1 & + & 7x_2 & + & x_3 & = & 19 \\ x_1 & - & 3x_2 & + & 12x_3 & = & 31 \end{cases} \tag{8}$$

Soluţie: Scriem formulele de recurenţă

$$\begin{cases} x_1^{(k+1)} & = & & - & 1/4x_2^{(k)} & + & 1/4x_3^{(k)} & + & 3/4 \\ x_2^{(k+1)} & = & -2/7x_1^{(k)} & & & - & 1/7x_3^{(k)} & + & 19/7 \\ x_3^{(k+1)} & = & -1/12x_1^{(k)} & + & 1/4x_2^{(k)} & & & + & 31/12 \end{cases} \tag{9}$$

Dacă alegem $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0$ obţinem următoarele rezultate:

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_2^{(k)}$
0	0	0	0
1	0.75	2.50	3.15
2	0.91	2.00	3.01
3	1.00	2.00	3.00
4	1.00	2.00	3.00

(10)

Soluţia exactă este: $x_1 = 1,\ x_2 = 2,\ x_3 = 3$.

2. Folosiţi metoda Jacobi pentru a aproxima soluţia sistemului:

$$\begin{cases} 5x_1 & - & 2x_2 & + & 3x_3 & = & -1 \\ -3x_1 & + & 9x_2 & + & x_3 & = & 2 \\ 2x_1 & - & x_2 & - & 7x_3 & = & 3 \end{cases} \tag{11}$$

Soluţie: Scriem formulele de recurenţă

$$\begin{cases} x_1 & = & -1/5 & + & 2/5x_2 & - & 3/5x_3 \\ x_2 & = & 2/9 & + & 3/9x_2 & - & 1/9x_3 \\ x_3 & = & -3/7 & + & 2/7x_2 & - & 1/7x_3 \end{cases} \tag{12}$$

Alegând $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0 \Rightarrow$

n	0	1	2	3	4	5	6	7
x_1	0.000	−0.200	0.146	0.192	0.181	0.185	0.186	0.186
x_2	0.000	0.222	0.203	0.328	0.332	0.329	0.331	0.331
x_3	0.000	−0.429	−0.517	−0.416	−0.421	−0.424	−0.423	−0.423

(13)

3. Fie sistemul $Ax = b, A \in R^{2\times 2}, x, b \in R^2, A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$. Matricea A nu este diagonal dominantă pe linii. În aceste condiţii este convergentă metoda Gauss-Seidel?
   Soluţie: Se determină matricea de iteraţie a sistemului pentru metoda Gauss-Seidel, G_{GS}.
   $A = D - L - U \Rightarrow \quad D = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, \quad L = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix}$
   Atunci:
   $G_{GS} = (D-L)^{-1}U = \begin{bmatrix} 3 & 0 \\ -1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix} == \frac{1}{6} * \begin{bmatrix} 3 & 0 \\ -1 & 2 \end{bmatrix} * \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & \frac{1}{3} \end{bmatrix}.$
   $\det(\lambda I - G_{GS}) = \begin{vmatrix} \lambda & 1 \\ 0 & \lambda - \frac{1}{3} \end{vmatrix} = 0 \Rightarrow \lambda(G_{GS}) = \{0, \frac{1}{3}\}$ şi $\rho(G_{GS}) = \frac{1}{3} < 1$.
   $\Rightarrow$ metoda Gauss-Seidel este convergentă.

Probleme Propuse

Problema 1

Matricea sistemului liniar

$$\begin{cases} 2x + y + z = 4 \\ x + 2y + z = 4 \\ x + y + 2z = 4 \end{cases} \tag{14}$$

nu este strict diagonal dominantă. În aceste condiţii este convergentă:

1. Metoda Jacobi
2. Metoda Gauss-Seidel ?

Dacă da, calculaţi soluţia iterativă la pasul 3. Alegeţi voi aproximaţia iniţială.

Problema 2

Fie sistemul $Ax = b, A \in R^{2\times 2}, b \in R^2$, cu $A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. Determinaţi raza spectrală a matricei de iteraţie Jacobi.

Problema 3

Fie o matrice $A \in R^{n \times n}$ tridiagonală [1] şi sistemul de ecuaţii

$Ax = b$, cu $b,\ x \in R^n$.

Scrieţi o funcţie OCTAVE care rezolvă sistemul de ecuaţii prin metoda iterativă Jacobi.

```
function x = solJacobi(A,b,x0,tol,maxiter)
  % Rezolvarea sistemului Ax=b folosind metoda Jacobi
  % Intrari:
  %   A - matricea sistemului
  %   b - vectorul termenilor liberi
  %   x0 - aproximatia intiala a solutiei
  %   tol - precizia determinarii solutiei
  %   maxiter - numarul maxim de iteratii permis
  % Iesiri:
  %   x - solutia sistemului

  % code goes here

endfunction
```

Listing 1: solJacobi.m

Problema 4

Să se implementeze o funcţie OCTAVE care rezolvă un sistem liniar de ecuaţii folosind metoda iterativă Gauss-Seidel. Funcţia va avea semnătura:

```
function x = solGaussSeidel(A,b,x0,tol,maxiter)
  % Rezolvare sistem cu Gauss-Seidel
  % Intrari:
  %   A - matricea sistemului
  %   b - vectorul termenilor liberi
  %   x0 - aproximatia intiala a solutiei
  %   tol - precizia determinarii solutiei
  %   maxiter - numarul maxim de iteratii permis
  % Iesiri:
  %   x - solutia sistemului

  % code goes here

endfunction
```

Listing 2: solGaussSeidel.m

[1] http://en.wikipedia.org/wiki/Tridiagonal_matrix

Problema 5

Să se implementeze o funcţie OCTAVE care rezolvă un sistem liniar de ecuaţii folosind metoda suprarelaxării. Funcţia va avea următoarea semnătură:

```
function [ x, flag ] = sor(A, x, b, w, max_it, tol)
  %
  % [ x, flag ] = sor(A, x, b, w, max_it, tol)
  %
  %   Metoda Suprarelaxarii
  %   Functia rezolva sisteme liniare Ax=b folosind metoda suprarelaxarii
  % Input:
  %   A - matricea sistemului
  %   x - aproximarea intiala a sistemului
  %   b - vectorul termenilor liberi
  %   w - factorul de relaxare
  %   max_it - numarul maxim de iteratii
  %   tol - toleranta
  % Output:
  %   x - solutia sistemului
  %   flag - 0 = a fost gasita o solutie / 1 = metoda nu converge pentru max_it

  % code goes here

endfunction
```

Listing 3: sor.m

Să se testeze funcţia folosind diferite valori pentru ω.