Metode iterative pentru rezolvarea sistemelor de ecuații liniare: Jacobi, Gauss-Siedel, Suprarelaxare

Colaboratori: Andrei STAN, Bogdan Țigănoaia

March 8, 2025

Cuprins

1 Obiective laborator

În urma parcurgerii acestui laborator, studentul va fi capabil să rezolve sisteme de ecuaţii liniare utilizând metode iterative.

2 Noțiuni teoretice

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.

Metodele iterative transformă sistemul $Ax = b$ în $x = Gx + c$. Pornindu-se cu o aproximaţie iniţială $x^{(0)}$ a soluţiei, relaţia de recurenţă folosită are forma:

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

unde:

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

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

$$\rho(G) < 1$$

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:

$(N - P)x = b$, adică $x = N^{-1}Px + N^{-1}b$.

Astfel, rezultă relaţia de recurenţă:

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

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:

$$A = D - L - U.$$

2.1 Metoda Jacobi

În metoda Jacobi se aleg:

$N = D$

$P = L + U$

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

Soluţia sistemului este:

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

2.2 Metoda Gauss-Seidel

La această metodă se aleg:

$N = D - L$

$P = U$

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

Soluţia sistemului este:

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

Observaţii:

1. Dacă matricea sistemului este diagonal dominantă pe linii, metoda Gauss Seidel este convergentă. Reciproca nu este adevarată.
2. O matrice A este diagonal dominantă pe linii dacă şi numai dacă are următoarea proprietate: pentru fiecare linie i, modulul elementului de pe diagonala principală, $A(i, i)$ este strict mai mare decât suma modulelor elementelor de pe aceeaşi linie i.

2.3 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:

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

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

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

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:

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

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

3 Probleme rezolvate

3.1 Problema 1

Să se rezolve sistemul folosind metoda Gauss-Seidel:

$$\begin{cases} 7x_1 + 2x_2 - 4x_3 = 7 \\ 3x_1 + 6x_2 + 2x_3 = 15 \\ 2x_1 - 5x_2 + 8x_3 = 28 \end{cases}$$

Soluţie:

Scriem formulele de recurenţă

$$\begin{cases} x_1^{(k+1)} = -2/7x_2^{(k)} + 4/7x_3^{(k)} + 7/7 \\ x_2^{(k+1)} = -3/6x_1^{(k)} - 2/6x_3^{(k)} + 15/6 \\ x_3^{(k+1)} = -2/8x_1^{(k)} + 5/8x_2^{(k)} + 28/8 \end{cases}$$

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

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	1	2	4.5
2	3.00	−0.5	2.43
3	2.53	0.41	3.12
4	2.66	0.12	2.91
5	2.62	0.21	2.97

Soluţia exactă este: $x_1 = 2.63$, $x_2 = 0.19$ $x_3 = 2.96$.

3.2 Problema 2

Folosiţi metoda Jacobi pentru a aproxima soluţia sistemului:

$$\begin{cases} 10x_1 - 5x_2 + x_3 = 1 \\ x_1 + 4x_2 + 3x_3 = 4 \\ 4x_1 - 3x_2 - 9x_3 = 6 \end{cases}$$

Soluţie:

Scriem formulele de recurenţă

$$\begin{cases} x_1^{(k+1)} = 5/10x_2^{(k)} - 1/10x_3^{(k)} + 1/10 \\ x_2^{(k+1)} = -1/4x_1^{(k)} - 3/4x_3^{(k)} + 4/4 \\ x_3^{(k+1)} = 4/9x_1^{(k)} - 3/9x_2^{(k)} - 6/9 \end{cases}$$

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

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0	0	0
1	0.1	1.00	-0.66
2	0.66	1.47	-1.95
3	0.93	1.55	-0.86
4	0.96	1.41	-0.76
5	0.88	1.33	-0.71

Soluţia exactă este: $x_1 = 0.84,\ x_2 = 1.34,\ x_3 = -0.73$.

3.3 Problema 3

Fie sistemul $Ax = b, A \in R^{2\times2}, x, b \in R^2, A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$. 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:

$G_{GS} = (D - L)^{-1}U = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & \frac{1}{3} \end{bmatrix}$.

$\det(\lambda I - G_{GS}) = \begin{vmatrix} \lambda & 1 \\ 0 & \lambda - \frac{1}{3} \end{vmatrix} = 0 \Rightarrow \lambda(G_{GS}) = \{0, \frac{1}{3}\}$ şi $\rho(G_{GS}) = \frac{1}{3} < 1$.

$\Rightarrow$ metoda Gauss-Seidel este convergentă.

3.4 Problema 4

Să se implementeze o funcţie OCTAVE care rezolvă un sistem de ecuaţii liniare folosind metoda iterativă Gauss-Seidel. Date de intrare: A - matricea sistemului; b - vectorul termenilor liberi; $x0$ - aproximaţia iniţială a soluţiei; *tol* - precizia determinării soluţiei; *maxiter* - numărul maxim de iteraţii. Date de ieşire: x - soluţia sistemului; *succes* - variabilă care indică convergenţa metodei.

Soluţie:

```
function [x succes] = GaussSeidel(A, b, x0, tol, maxiter)
  [m n] = size(A);
  x = x0;
  succes = 0;

  while maxiter > 0
    maxiter--;
    xp = zeros(1, n);

    for i = 1 : m
      suma1 = 0;
      suma2 = 0;

      for j = 1 : i - 1
```

```
      suma1 += A(i, j)*xp(j);
    endfor

    for j = i + 1 : n
      suma1 += A(i, j)*x(j);
    endfor

    xp(i) = (b(i)-suma1-suma2)/A(i, i);
  endfor

  x = xp;
  if norm(xp-x, 2) < tol
    succes = 1;
    break;
  endif
 endwhile
endfunction
```

Date de intrare:	**Date de ieşire:**
$A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, x0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, tol = 0.0001, maxiter = 100$	$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

4 Probleme propuse

4.1 Problema 1

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. Stabiliţi convergenţa metodei Jacobi.

4.2 Problema 2

Fie sistemul liniar:

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

Stabiliţi:

a) dacă matricea sistemului este diagonal dominantă pe linii;

b) convergenţa metodei Jacobi;

c) convergenţa metodei Gauss-Seidel.

În caz de convergenţă, calculaţi soluţia iterativă după trei paşi. Alegeţi voi aproximaţia iniţială.

4.3 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.

[1] http://mathworld.wolfram.com/TridiagonalMatrix.html

```
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
```

Listing 1: Algoritmul Jacobi

4.4 Problema 4

Să se implementeze o funcţie OCTAVE care rezolvă un sistem liniar de ecuaţii folosind metoda suprarelaxării.

```
function [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
```

Listing 2: Suprarelaxare

Să se testeze funcţia folosind diferite valori pentru ω.