Lucrarea de laborator nr. 6

I. Scopul lucrării

Metode iterative de rezolvare a sistemelor liniare.

II. Conținutul lucrării

1. Metode iterative de rezolvare a sistemelor liniare. Generalități.
2. Metoda Jacobi.
3. Metoda Gauss-Seidel.

III. Prezentarea lucrării

III.1. Metode iterative de rezolvare a sistemelor liniare. Generalități.

Metodele iterative constau în construcția unui şir $(x^k)_k$ convergent către soluția exactă a sistemului. Oprirea procesului iterativ are loc la un indice m determinat pe parcursul calculului în funcție de precizia impusă astfel încât termenul x^m să constituie o aproximație satisfăcătoare a soluției căutate.

Se consideră sistemul liniar

$$(1) \quad Ax = b,\ A \in M_{n,n}(\mathbf{R}) \text{ nesingulară}$$

şi ***desfacerea matricei*** A definită prin

$$A = N\text{-}P.$$

cu N o matrice inversabilă.
Fie x^0 un vector arbitrar din $\mathbf{R}^n$. Construim şirul $(x^k)_k$ folosind relația de recurență:

$$(2) \quad Nx^{k+1} = P\,x^k + b,\ k \geq 0.$$

Fie $G = N^{-1}P$. Este cunoscut că şirul $(x^k)_k$ converge la soluția exactă a sistemului oricare ar fi x^0 dacă şi numai dacă $\rho(G) < 1$ ($\rho(G)$ reprezintă raza spectrală a lui G, i.e. maximum modulelor valorilor proprii ale lui G). Fie x soluția exactă a sistemului. Dacă notăm eroarea lui x^k față de x cu

$$e^k = x - x^k,$$

atunci $e^{k+1} = G\,e^k = G^k e^0$, pentru orice $k \geq 0$.

N se alege astfel încât sistemul (2) să se rezolve uşor – de exemplu diagonală sau triunghiulară În cazul metodelor concrete descrise în continuare se consideră **desfacerea standard** a matricei A = $(a_{i,j})_{1\le i,j\le n}$ definită prin:

$$A = L + D + R$$

unde

$$D = \mathrm{diag}(a_{1,1},a_{2,2},\ldots a_{n,n}) = \begin{pmatrix} a_{1,1} & 0 & 0 & \ldots\ldots & 0 & 0 \\ 0 & a_{2,2} & 0 & \ldots\ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots\ldots & 0 & a_{n,n} \end{pmatrix}$$

$$L = \begin{pmatrix} 0 & 0 & 0 & \ldots\ldots & 0 & 0 \\ a_{2,1} & 0 & 0 & \ldots\ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n,1} & a_{n,2} & a_{n,3} & \ldots\ldots & a_{n,n-1} & 0 \end{pmatrix}$$

$$R = \begin{pmatrix} 0 & a_{1,2} & a_{1,3} & \ldots\ldots & a_{1,n-1} & a_{1,n} \\ 0 & 0 & a_{2,3} & \ldots\ldots & a_{2,n-1} & a_{2,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots\ldots & 0 & 0 \end{pmatrix}$$

III.2. Metoda Jacobi

Metoda Jacobi se caracterizează prin desfacerea

$$N = D,\ P = -(L+R)$$

Şirul $(x^k)_k$ construit prin această metodă este:

$$x_i^{k+1} = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j} x_j^k}{a_{i,i}},\ i = 1,2,\ldots,n$$

Dacă $G = N^{-1}P$, atunci coeficienții matricei G sunt:

$$g_{i,j} = \begin{cases} 0, & i = j \\ -\dfrac{a_{i,j}}{a_{i,i}}, & i \neq j. \end{cases}$$

$\rho(G) \le \min(\|G\|_1, \|G\|_\infty)$.

Calculăm $\|G\|_1$:

$$\|G\|_1 = \max\{\sum_{i=1}^{n}|g_{i,j}|, 1 \le j \le n\}$$

$$= \max\{\sum_{\substack{i=1\\ i\neq j}}^{n}\frac{|a_{i,j}|}{|a_{i,i}|},\ 1 \le j \le n\}$$

Calculăm $\|G\|_\infty$:

$$\|G\|_\infty = \max\{\sum_{j=1}^{n}|g_{i,j}|, 1 \le i \le n\}$$

$$= \max\{\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{|a_{i,j}|}{|a_{i,i}|}, 1 \le i \le n\}.$$

Dacă $\|G\|_1 < 1$ sau $\|G\|_\infty < 1$, atunci $\rho(G) < 1$. Dar condiția $\|G\|_\infty < 1$ este echivalentă cu:

$$|a_{i,i}| > \sum_{\substack{i=1\\ i\neq j}}^{n}|a_{i,j}|$$

caz în care spunem că A este ***diagonal dominantă***. Deci dacă $a_{i,i} \neq 0$ pentru orice i = 1,2,…, n, şi dacă A este diagonal dominantă atunci şirul obținut prin metoda Jacobi converge la soluția exactă a sistemului (1). Dacă e^k este eroarea lui x^k față de x:

$$e^k = x - x^k,$$

atunci $e^{k+1} = G\, e^k = G^k e^0$, pentru orice $k \ge 0$. În consecință pentru orice norme compatibile

$$\|e^k\| \le \|G^k e^0\| \le \|G^k\|\|e^0\| \le \|G\|^k\|e^0\|$$

Fie $n_{1,\infty} = \min(\|G\|_1, \|G\|_\infty)$ şi fie eps > 0 dat. Vom considera x^k este o aproximație satisfăcătoare pentru soluția exactă a sistemului dacă

$$(n_{1,\infty})^k < eps.$$

ceea ce este echivalent cu

$$k \ge \left[\frac{\ln(eps)}{\ln(n_{1,\infty})}\right] + 1.$$

Următoarea procedură Maple are drept parametri numărul de ecuații ale sistemului, matricea extinsă a sistemului, aproximația inițială x^0 a soluției, şi eroarea eps. Procedura întoarce aproximația x^k (dată de metoda Jacobi) a soluției cu

$$k = \left[\frac{\ln(eps)}{\ln(n_{1,\infty})}\right] + 1.$$

```
> mjacobi := proc(n, a, x0, eps)
    local n1, ni, x, k, i, j, p, suma;
    x := vector(n);
    ni := 0;
    for i to n do
        suma := 0;
        for j to i - 1 do
         suma := suma + abs(a[i, j])
       od;
        for j from i + 1 to n do
         suma := suma + abs(a[i, j])
        od;
        if a[i, i] = 0 then
         print(`Metoda nu se aplica`);
         RETURN
        fi;
        suma := suma/abs(a[i, i]);
        if ni < suma then
         ni := suma
       fi
      od;
      n1 := 0;
      for j to n do
       suma := 0;
   for i to j - 1 do
    suma := suma + abs(a[i, j] /a[i,i])
   od;
   for i from j + 1 to n do
    suma := suma + abs(a[i, j]/a[i,i])
   od;
   if n1 < suma then
    n1 := suma fi
   od;
   if n1 < ni then
     ni := n1
```

```
   fi;
   if 1 <= ni then
    print(`Metoda nu se aplica`);
    RETURN
   fi;
   k := floor(ln(eps)/ln(ni)) + 1;
   for p to k do
           for i to n do
            x[i] := a[i, n + 1];
            for j to i - 1 do
x[i] := x[i] - a[i, j]*x0[j]
            od;
            for j from i + 1 to n do
x[i] := x[i] - a[i, j]*x0[j]
            od;
            x[i] := x[i]/a[i, i]
          od;
          x0 := x
        od;
        RETURN(evalm(x))
end;
```

Utilizăm această procedură pentru rezolvarea sistemului:

$$(3)\quad \begin{cases} 3x_1 + x_2 + x_3 = 5 \\ x_1 + 5x_2 + x_3 = 7 \\ -x_1 + x_2 + 8x_3 = 8 \end{cases}$$

```
>A:=matrix(3,4,[3,1,1,5,1,5,1,7,-1,1,8,8]);
```

$$A := \begin{pmatrix} 3 & 1 & 1 & 5 \\ 1 & 5 & 1 & 7 \\ -1 & 1 & 8 & 8 \end{pmatrix}$$

```
> x0:=vector(3,[0,0,0.1]);
                    x0 := [0, 0, .1]
> mjacobi (3,a,x0,0.001);
      [.9999999999, 1.000000000, 1.000000000]
> x1:=vector(3,[0.,0.1,2345]);
                           x1 := [0, .1, 2345]
> mjacobi (3,a,x1,10^(-7));
      [.9999999999, 1.000000000, 1.000000000]
```

III.3. Metoda Gauss-Seidel.

Metoda Gauss Seidel corespunde desfacerii

$$N = L + D,\ P = -R.$$

Şirul $(x^k)_k$ construit prin această metodă este dat prin:

$$x_1^{k+1} = \frac{b_1 - \sum_{j=2}^{n} a_{1,j} x_j^k}{a_{1,1}},$$

$$x_i^{k+1} = \frac{b_i - \sum_{j=i+1}^{n} a_{i,j} x_j^k - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1}}{a_{i,i}},\ i = 2,3,\dots,n$$

Ca şi în cazul metodei Jacobi dacă $a_{i,i} \neq 0$ pentru orice i = 1,2,…, n, şi dacă A este diagonal dominantă atunci şirul obţinut prin metoda Gauss-Seidel converge la soluţia exactă a sistemului (1). De asemenea dacă A este o matrice simetrică şi are elementele de pe diagonala principală pozitive, atunci metoda Gauss-Seidel converge dacă şi numai dacă matricea A este pozitiv definită.

Considerăm A diagonal dominantă şi folosim aceeaşi evaluare de eroare ca la metoda Jacobi. Procedura Maple de mai jos are drept parametri numărul de ecuaţii ale sistemului, matricea extinsă a sistemului, aproximaţia iniţială x^0 a soluţiei, şi eroarea eps. Procedura întoarce aproximaţia x^k (dată de metoda Gauss.Seidel) a soluţiei cu

$$k = \left[\frac{\ln(eps)}{\ln(n_{1,\infty})} \right] + 1.$$

```
> mgaussseidel := proc(n, a, x0, eps)
 local n1, ni, x, k, i, j, p, suma;
 x := vector(n);
 ni := 0;
 for i to n do
 suma := 0;
 for j to i - 1 do
   suma := suma + abs(a[i, j])
 od;
 for j from i + 1 to n do
   suma := suma + abs(a[i, j])
  od;
```

```
  if a[i, i] = 0 then
   print(`Metoda nu se aplica`); RETURN
  fi;
  suma := suma/abs(a[i, i]);
  if ni < suma then ni := suma fi
 od;
 n1 := 0;
 for j to n do
   suma := 0;
   for i to j - 1 do
    suma := suma + abs(a[i, j])
   od;
   for i from j + 1 to n do
     suma := suma + abs(a[i, j])
   od;
   suma := suma/abs(a[j, j]);
   if n1 < suma then n1 := suma fi
  od;
  if n1 < ni then ni := n1 fi;
  if 1 <= ni then
   print(`Metoda nu se aplica`); RETURN
  fi;
  k := floor(ln(eps)/ln(ni)) + 1;
  for p to k do
    x[1] := a[1, n + 1];
    for j from 2 to n do
     x[1] := x[1] - a[1, j]*x0[j]
   od;
   x[1] := x[1]/a[1, 1];
   for i from 2 to n do
     x[i] := a[i, n + 1];
   for j from i + 1 to n do
     x[i] := x[i] - a[i, j]*x0[j]
   od;
   for j to i - 1 do
    x[i] := x[i] - a[i, j]*x[j]
   od;
   x[i] := x[i]/a[i, i]
  od;
  x0 := x
  od;
  RETURN(evalm(x))
end;
```

Aplicăm această metodă pentru rezolvarea sistemului (3) din secțiunea precedentă.

```
>A:=matrix(3,4,[3,1,1,5,1,5,1,7,-1,1,8,8]);
```

$$A := \begin{bmatrix} 3 & 1 & 1 & 5 \\ 1 & 5 & 1 & 7 \\ -1 & 1 & 8 & 8 \end{bmatrix}$$

```
> x0:=vector(3,[0,0,0.1]);
                  x0 := [0, 0, .1]
> mgaussseidel(3,a,x0,0.001);
      [.9999999999, 1.000000000, 1.000000000]
> x1:=vector(3,[0.,0.1,2345]);
                 x1 := [0, .1, 2345]
> mgaussseidel(3,a,x1,10^(-7));
      [.9999999999, 1.000000000, 1.000000000].
```

Probleme propuse

1) Analizați comparativ metodele Jacobi și Gauss-Seidel pentru sistemul Ax = b, cu

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad x^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

2) Aceeași problemă pentru

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$