Rezolvarea sistemelor de ecuaţii liniare

Laurenţiu DEACONU*

Abstract

Ne propunem să realizăm o aplicaţie (un program) care să rezolve un sistem de ecuaţii liniare. Nu insistăm asupra sistemelor de ecuaţii liniare compatibile determinate (pentru care rangul matricei sistemului este egal cu rangul matricei extinse şi cu numărul de necunoscute) pentru care există multe metode simple de rezolvare (inclusiv metoda Crammer) care se pot transfera în algoritmi informatici, ci prezentăm o metodă care permite atât stabilirea compatibilităţii cât şi rezolvarea sistemelor compatibile nedeterminate.

1 Argumentare matematică

Considerăm sistemul de ecuaţii liniare

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}, \tag{1}$$

cu $a_{ij}, b_i \in \mathbb{R}$, $i \in \overline{1,m}, j \in \overline{1,n}$, $m, n \in \mathbb{N}^\star$.

Algoritmul folosit pentru rezolvarea sistemului are la bază lema substituţiei. Pentru simplificarea calculelor vom organiza rezolvarea transferând sistemul într-un tabel. În mod normal, dacă sistemul este compatibil, algoritmul are m paşi. La pasul k ($0 \le k \le m$) tabelul are următoarea structură:

*Universitatea din Piteşti, laurentiu.deaconu@upit.ro

$$
\begin{array}{c|cccccc|c}
 & x_1 & x_2 & \dots & x_l & \dots & x_n & \\
\cline{2-7}
 & j_1^{[k]} & j_2^{[k]} & \dots & j_l^{[k]} & \dots & j_n^{[k]} & \\
\hline
i_1^{[k]} & a_{11}^{[k]} & a_{12}^{[k]} & \dots & a_{1l}^{[k]} & \dots & a_{1n}^{[k]} & b_1^{[k]} \\
i_2^{[k]} & a_{21}^{[k]} & a_{22}^{[k]} & \dots & a_{2l}^{[k]} & \dots & a_{2n}^{[k]} & b_2^{[k]} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
i_k^{[k]} & a_{k1}^{[k]} & a_{k2}^{[k]} & \dots & a_{kl}^{[k]} & \dots & a_{kn}^{[k]} & b_k^{[k]} \\
0 & a_{k+1,1}^{[k]} & a_{k+1,2}^{[k]} & \dots & a_{k+1,l}^{[k]} & \dots & a_{k+1,n}^{[k]} & b_{k+1}^{[k]} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & a_{m'1}^{[k]} & a_{m'2}^{[k]} & \dots & a_{m'l}^{[k]} & \dots & a_{m'n}^{[k]} & b_{m'}^{[k]}
\end{array}
$$

Elementele tabelului vor fi precizate în cadrul algoritmului.

Pasul 1° Generarea tabelului. Primul tabel se completează astfel:

1. Prima linie $x_1, x_2, \dots, x_n$ este doar o notaţie pentru coloanele asociate necunoscutelor sistemului. Ea nu se modifică pe parcursul algoritmului.

2. A doua linie $j_1, j_2, \dots, j_n$ se completează cu 0, $j_l = 0$, $\forall j \in \overline{1, n}$. Ea se completează cu elemente nenule la fiecare pas.

3. Coloana din stânga cu elementele $i_1, i_2, \dots, i_{m'}$ are iniţial toate elementele nule, $i_k = 0$, $\forall k \in \overline{1, m'}$. La primul pas avem $m' = m$. Se completează cu elemente nenule la fiecare pas al algoritmului.

4. Zona centrală conţine elementele matricei sistemului. La primul tabel ($k = 0$) avem $a_{ij}^{[0]} = a_{ij}$, $\forall i \in \overline{1, m}$, $\forall j \in \overline{1, n}$.

5. Coloana din dreapta se completează cu termenii liberi ai sistemului. Avem $b_i^{[0]} = b_i$, $\forall i \in \overline{1, m}$.

Pasul 2° Prelucrarea tabelului. La pasul k, ($k \in \overline{1, m'}$), parcurgem următoarele etape:

1. Căutăm, în linia $a_{k1}^{[k-1]}, a_{k2}^{[k-1]}, \dots, a_{kl}^{[k-1]}, \dots, a_{kn}^{[k-1]}$ din zona centrală, un element nenul. Dacă găsim, îl numim **pivot**, identificăm coloana în care se află (notată x_l în tabelul de mai sus) şi trecem la pasul următor. Dacă avem $a_{kq}^{[k-1]} = 0$, $\forall q \in \overline{1, n}$, atunci:

   (a) dacă $b_k^{[k-1]} = 0$ eliminăm linia k deoarece corespunde unei ecuaţii care este o combinaţie liniară a ecuaţiilor precedente şi care nu contează în rezolvarea sistemului. În acest caz numărul ecuaţiilor se micşorează cu 1, $m' \leftarrow m' - 1$. Dacă avem $k \leq m'$, reluăm etapa 1, altfel trecem la Pasul 3°.

(b) dacă $b_k^{[k-1]} \neq 0$ algoritmul se opreşte. **Sistemul este incompatibil.**

2. În coloana din stânga, punem $i_k^{[k]} = l$, unde x_l este coloana în care a fost identificat pivotul. Celelalte elemente rămân ca la pasul precedent, $i_p^{[k]} = i_p^{[k-1]}$, $\forall p \in \{1, 2, \dots, m'\} \setminus \{k\}$.

3. În linia a doua, punem $j_l^{[k]} = k$ şi $j_q^{[k]} = j_q^{[k-1]}$, $\forall q \in \{1, 2, \dots, n\} \setminus \{l\}$.

4. Elementele din linia care conţine pivotul se împart la pivot:

$$a_{kq}^{[k]} = \frac{a_{kq}^{[k-1]}}{a_{kl}^{[k-1]}}, \quad \forall q \in \overline{1, n}, \quad b_k^{[k]} = \frac{b_k^{[k-1]}}{a_{kl}^{[k-1]}}.$$

   Evident $a_{kl}^{[k]} = 1$.

5. Celelalte elemente din coloana x_l, în care se află pivotul, vor fi egale cu 0, $a_{pl}^{[k]} = 0$, $\forall p \in \{1, 2, \dots, m'\} \setminus \{k\}$.

6. Toate elementele care nu au fost înlocuite în etapele 2-5 se determină folosind *regula dreptunghiului*:

$$a_{pq}^{[k]} = \frac{a_{pq}^{[k-1]} a_{kl}^{[k-1]} - a_{kq}^{[k-1]} a_{pl}^{[k-1]}}{a_{kl}^{[k-1]}} = a_{pq}^{[k-1]} - \frac{a_{kq}^{[k-1]} a_{pl}^{[k-1]}}{a_{kl}^{[k-1]}},$$

   $\forall p \in \{1, 2, \dots, m'\} \setminus \{k\}$, $\forall q \in \{1, 2, \dots, n\} \setminus \{l\}$.

7. Dacă $k < m'$, creştem valoarea lui k cu 1 şi reluăm Pasul 2°, altfel trecem la Pasul 3°.

Pasul 3° Recuperarea soluţiei. **Sistemul este compatibil**:

1. **determinat**, dacă $m' = n$. Soluţia este $x_l = b_{j_l^{[m']}}^{[m']}$, $1 \leq l \leq n$.

2. **nedeterminat**, dacă $m' < n$. Construim mulţimea $S = \{l \in \mathbb{N} \mid j_l^{[m']} = 0, 1 \leq l \leq n\}$. Parcurgem a doua linie, $j_1^{[m']}, j_2^{[m']}, \dots, j_n^{[m']}$:

   (a) dacă $j_l^{[m']} = 0$, $1 \leq l \leq n$, atunci x_l este necunoscută secundară şi avem $x_l = \alpha_l$, unde $\alpha_l \in \mathbb{R}$ este un parametru.

   (b) dacă $j_l^{[m']} \neq 0$, $1 \leq l \leq n$, atunci $x_l = b_{j_l^{[m']}}^{[m']} - \sum_{q \in S} a_{j_l^{[m']} q}^{[m']} \alpha_q$.

2 Exemplu de calcul

Considerăm sistemul

$$\begin{cases} x_1+2x_2+3x_3+\ 4x_4+\ 5x_5 = \ 6 \\ 2x_1+\ x_2+3x_3+\ 2x_4+\ \ x_5 = \ 8 \\ x_1+5x_2+6x_3+10x_4+14x_5 = 10 \\ 2x_1+\ x_2+6x_3+\ 5x_4+\ 4x_5 = \ 3 \end{cases}, \tag{2}$$

Pentru rezolvare utilizăm algoritmul prezentat mai sus. Mai întâi scriem tabelul iniţial.

	x_1	x_2	x_3	x_4	x_5	
	0	0	0	0	0	
0	$\boxed{1}$	2	3	4	5	6
0	2	1	3	2	1	8
0	1	5	6	10	14	10
0	2	1	6	5	4	3

În linia corespunzătoare primei ecuaţii toate elementele sunt nenule, deci oricare element poate fi ales ca *pivot*. Pentru a evita calculele complicate, alegem ca pivot primul element, pe 1, elementul încadrat într-un pătrat în tabelul anterior. Prelucrăm primul tabel:
– în coloana din stânga, în linia pivotului, punem 1 deoarece am ales *pivotul* în coloana lui x_1; restul elementelor se transcriu din primul tabel;
– în linia de deasupra zonei centrale, în coloana *pivotului*, sub x_1, punem 1 deoarece *pivotul* se găseşte în linia 1 din zona centrală, linia corespunzătoare primei ecuaţii; restul elementelor se transcriu din primul tabel;
– linia *pivotului* se împarte la *pivot*, adică se transcrie deoarece *pivotul* este egal cu 1;
– coloana *pivotului* se completează cu zero-uri;
– restul elementelor se calculează cu *regula dreptunghiului*; de exemplu, elementul din linia 4 şi coloana 3 din zona centrală (care este 6 în primul tabel) se calculează astfel:

$$a_{43}^{[1]} = \frac{6 \cdot 1 - 2 \cdot 3}{1} = 0,$$

6 este elementul corespunzător din tabelul anterior, 1 este *pivotul*, 2 şi 3 sunt "proiecţiile" elementului înlocuit pe coloana, respectiv linia, *pivotului*. Termenii liberi se calculează în acelaşi mod: elementul din linia 3 din coloana termenilor liberi se calculează astfel

$$b_3^{[1]} = \frac{10 \cdot 1 - (1) \cdot 6}{1} = 4.$$

Obţinem tabelul următor:

	x_1	x_2	x_3	x_4	x_5	
	1	0	0	0	0	
1	1	2	3	4	5	6
0	0	$\boxed{-3}$	-3	-6	-9	-4
0	0	3	3	6	9	4
0	0	-3	0	-3	-6	-9

În linia corespunzătoare ecuaţiei 2 alegem pivotul -3 din coloana lui x_2. În coloana din stânga, în linia pivotului punem 2 deoarece *pivotul* este în coloana lui x_2, iar în linia de deasupra zonei centrale, sub x_2, punem 2 deoarece *pivotul* se găseşte în linia 2 din zona centrală. Restul elementelor se completează ca la pasul precedent. Obţinem tabelul:

	x_1	x_2	x_3	x_4	x_5	
	1	2	0	0	0	
1	1	0	1	0	-1	$\frac{10}{3}$
2	0	1	1	2	3	$\frac{4}{3}$
0	0	0	0	0	0	0
0	0	0	3	3	3	-5

Observăm că în linia 3 din zona centrala, în coloanele corepunzătoare necunoscutelor toate elementele sunt egale cu 0, deci nu putem alege un *pivot*. Elementul corespunzător din coloana termenilor liberi este tot 0, deci putem renunţa la ecuaţia 3. Eliminăm această linie şi obţinem tabelul:

	x_1	x_2	x_3	x_4	x_5	
	1	2	0	0	0	
1	1	0	1	0	-1	$\frac{10}{3}$
2	0	1	1	2	3	$\frac{4}{3}$
0	0	0	$\boxed{3}$	3	3	-5

În tabelul anterior alegem *pivotul* 3 din ultima linie şi coloana lui x_3.

	x_1	x_2	x_3	x_4	x_5	
	1	2	3	0	0	
1	1	0	0	-1	-2	5
2	0	1	0	1	2	3
3	0	0	1	1	1	$-\frac{5}{3}$

Analizând ultimul tabel deducem că sistemul este compatibil nedeterminat. Necunoscutele principale sunt x_1, x_2, x_3, iar necunoscutele secundare x_4 şi x_5. Asociem paramatri reali necunoscutelor secundare: $x_4 = \alpha$, $x_5 = \beta$.

Pentru simplificarea identificării soluţiei putem completa tabelul anterior astfel:

				α	β	
	x_1	x_2	x_3	x_4	x_5	
	1	2	3	0	0	
1	1	0	0	-1	-2	5
2	0	1	0	1	2	3
3	0	0	1	1	1	$-\frac{5}{3}$

Pentru x_1, în linia de deasupra zonei centrale găsim 1. În coloana din stânga identificăm linia în care se găseşte elementul 1. Din această linie recuperăm x_1: din elementul din coloana termenilor liberi, 5, se scad produsele dintre elementele din coloanele corespunzătoare parametrilor şi respectivii parametri, adică $-1 \cdot \alpha$ şi $-2 \cdot \beta$. Procedăm similar pentru necunoscutele x_2 şi x_3. Obţinem astfel soluţia:

$$\begin{cases} x_1 = & 5+\alpha+2\beta \\ x_2 = & 3-\alpha-2\beta \\ x_3 = & -\frac{5}{3}-\alpha\ \ -\beta \\ x_4 = & \alpha \\ x_5 = & \beta \end{cases}, \tag{3}$$

3 Implementare în limbajul C++

Algorimul prezentat în acest articol permite realizarea unui program care

- determină dacă un sistem de ecuaţii liniare este compatibil sau incompatibil
- identifică dacă un sistem compatibil este compatibil determinat sau compatibil nedeterminat
- pentru sistemele compatibile determinate calculează soluţia
- pentru sistemele compatibile nedeterminate determină soluţia generală, în funcţie de parametri.

Programul prezentat mai jos, realizat în **Code::Blocks**, citeşte datele din fişierul text `sistem.txt` şi scrie rezultatele în fişierul text `solutie.txt`.

Fişierul prezentat mai jos (`sistem.txt`) conţine datele corespunzătoare sistemului (2). Prima linie conţine m, n, numărul ecuaţiilor, respectiv numărul necunoscutelor. În continuare se scriu elementele matricei extinse de dimensiune $m \times (n+1)$, ultima coloană conţinând termenii liberi.

```
4 5
 1   2   3   4   5   6
 2   1   3   2   1   8
 1   5   6  10  14  10
 2   1   6   5   4   3
```

Programul respectă algoritmul prezentat cu mici diferenţe:

– Pentru alegerea pivotului se determină elementul din linia curentă care are valoarea absolută maximă. Dacă acesta este nul, deducem că toate elementele liniei sunt nule. Se analizează Pasul 2°, etapa 1 (a) şi (b). Dacă elementul determinat este nenul, acesta va fi pivotul (se alege elementul cu valoare absolută maximă pentru reducerea erorilor de rotunjire).

– Pentru a verifica dacă un număr real este nul, nu îl comparăm cu 0 ci verificăm dacă valoarea absolută a lui este mai mică decât ε, unde ε este un număr real pozitiv foarte mic. În program $\varepsilon = 10^{-20}$.

– Prelucrarea tabelului din Pasul 2° se face parcurgând etapele în ordinea 6, 4, 5, 3, 2.

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <math.h>
using namespace std;

ifstream f1("sistem.txt");
ofstream f2("solutie.txt");
  int m,n,io,jo;
  double a[20][40],eps=1.0e-20;
  int o[40],v[20];

void citire()
{
  int i,j;
  f1>>m>>n;
  for(i=1;i<=m;i++)
  for(j=1;j<=n+1;j++)
    f1>>a[i][j];
  f1.close();
}

void generare_tabel()
{
  int i,j;
  for(i=1;i<=m;i++)
```

```
    v[i]=0;
  for(j=1;j<=n+1;j++)
    o[j]=0;
}

void prelucrare()
{
  int i,j;
  for(i=1;i<=m;i++)
  if(i!=io)
    for(j=1;j<=n+1;j++)
    if(j!=jo)
      a[i][j]=(a[i][j]*a[io][jo]-a[i][jo]*a[io][j])/a[io][jo];
  for(j=1;j<=n+1;j++)
  if(j!=jo)
    a[io][j]/=a[io][jo];
  for(i=1;i<=m;i++)
    a[i][jo]=0.0;
  a[io][jo]=1.0;
  o[jo]=io;
  v[io]=jo;
}

int pivot()
{
  int i,j;
  double max;
  max=fabs(a[io][1]);
  jo=1;
  for(j=2;j<=n;j++)
  if(max<=fabs(a[io][j]))
  {
    max=fabs(a[io][j]);
    jo=j;
  }
  if(max<eps)
  {
    if(fabs(a[io][n+1])<eps)
    {
      for(i=io;i<m;i++)
      for(j=1;j<=n+1;j++)
        a[i][j]=a[i+1][j];
      m--;
      return -1;
    }
    else
    {
      f2<<"Sistem incompatibil!";
      io=m+2;
      return 0;
    }
  }
  else
  {
```

```
    return 1;
  }
}

void recuperare_solutie()
{
  int j;
  char c[n];
  if(io==m+1)
  {
    if(m==n)
      f2<<"Sistemul este compatibil determinat."<<endl;
    else
      {
        f2<<"Sistemul este compatibil nedeterminat."<<endl;
        int k=96;
        for (j=1;j<=n;j++)
          if(o[j]==0)
            c[j]=char(++k);
      }
    f2<<"Solutia sistemului este:"<<endl;
    for(jo=1;jo<=n;jo++)
    {
      f2<<"x"<<noshowpos<<jo<<"="<<showpos;
      if(o[jo]==0)
      {
        f2<<c[jo]<<endl;
      }
      else
      {
        io=o[jo];
        f2<<setw(8)<<fixed<<setprecision(4)<<a[io][n+1];
        for(j=1;j<=n;j++)
        if((o[j]==0)&&(fabs(a[io][j]))>eps)
        {
          f2<<setw(8)<<fixed<<setprecision(4)<<-a[io][j]<<"*"<<c[j];
        }
        f2<<endl;
      }
    }
  }
}

void scrie_tabel()
{
  int i,j;
  f2<<"======";
  for(j=1;j<=n+1;j++)
    f2<<"========";
  f2<<endl;
  f2<<"  |";
  for(j=1;j<=n;j++)
    f2<<setw(5)<<o[j]<<"   ";
  f2<<"|"<<endl;
```

```
  f2<<"------";
  for(j=1;j<=n+1;j++)
    f2<<"--------";
  f2<<endl;
  for(i=1;i<=m;i++)
  {
    f2<<setw(4)<<v[i]<<"|";
    for(j=1;j<=n;j++)
      f2<<setw(8)<<fixed<<setprecision(3)<<a[i][j];
    f2<<"|"<<setw(8)<<fixed<<setprecision(3)<<a[i][n+1]<<endl;
  }
  f2<<"======";
  for(j=1;j<=n+1;j++)
    f2<<"========";
  f2<<endl;
}

int main(void)
{
  int p;
  citire();
  if (m==0||n==0)
  {
    cout<<"Sistemul nu exista!";
    return 1;
  }
  f2<<"Tabelul initial:"<<endl;
  generare_tabel();
  scrie_tabel();
  io=1;
  do
  {
    p=pivot();
    if(p==1)
    {
      prelucrare();
      f2<<"Pozitia pivotului: linia "<<io<<"  coloana "<<jo<<endl;
      scrie_tabel();
      io++;
    }
    else
    {
      if(p==-1)
      {
        f2<<"Eliminare ecuatie - linia "<<io<<endl;
        scrie_tabel();
      }
    }
  }
  while(io<=m);
  recuperare_solutie();
  f2.close();
  return 0;
}
```

Pentru sistemul (2) se obţine rezultatul din fişierul `solutie.txt`:

```
Tabelul initial:
==========================================================
    |    0        0        0        0        0   |
----------------------------------------------------------
   0|   1.000    2.000    3.000    4.000    5.000|   6.000
   0|   2.000    1.000    3.000    2.000    1.000|   8.000
   0|   1.000    5.000    6.000   10.000   14.000|  10.000
   0|   2.000    1.000    6.000    5.000    4.000|   3.000
==========================================================
Pozitia pivotului: linia 1  coloana 5
==========================================================
    |    0        0        0        0        1   |
----------------------------------------------------------
   5|   0.200    0.400    0.600    0.800    1.000|   1.200
   0|   1.800    0.600    2.400    1.200    0.000|   6.800
   0|  -1.800   -0.600   -2.400   -1.200    0.000|  -6.800
   0|   1.200   -0.600    3.600    1.800    0.000|  -1.800
==========================================================
Pozitia pivotului: linia 2  coloana 3
==========================================================
    |    0        0        2        0        1   |
----------------------------------------------------------
   5|  -0.250    0.250    0.000    0.500    1.000|  -0.500
   3|   0.750    0.250    1.000    0.500    0.000|   2.833
   0|   0.000    0.000    0.000    0.000    0.000|   0.000
   0|  -1.500   -1.500    0.000    0.000    0.000| -12.000
==========================================================
Eliminare ecuatie - linia 3
==========================================================
    |    0        0        2        0        1   |
----------------------------------------------------------
   5|  -0.250    0.250    0.000    0.500    1.000|  -0.500
   3|   0.750    0.250    1.000    0.500    0.000|   2.833
   0|  -1.500   -1.500    0.000    0.000    0.000| -12.000
==========================================================
Pozitia pivotului: linia 3  coloana 1
==========================================================
    |    3        0        2        0        1   |
----------------------------------------------------------
   5|   0.000    0.500   -0.000    0.500    1.000|   1.500
   3|   0.000   -0.500    1.000    0.500    0.000|  -3.167
   1|   1.000    1.000   -0.000   -0.000   -0.000|   8.000
==========================================================
Sistemul este compatibil nedeterminat.
Solutia sistemului este:
x1= +8.0000 -1.0000*a
x2=a
x3= -3.1667 +0.5000*a -0.5000*b
x4=b
x5= +1.5000 -0.5000*a -0.5000*b
```

Aşa cum se poate observa din fişierul cu rezultate, soluţia generală obţinută prin program pentru sistemul (2) este

$$\begin{cases} x_1 = & 8 & -a \\ x_2 = & a & \\ x_3 = & -\frac{19}{6} & +\frac{1}{2}a - \frac{1}{2}b \\ x_4 = & b & \\ x_5 = & \frac{3}{2} & -\frac{1}{2}a - \frac{1}{2}b \end{cases}, \tag{4}$$

În această formă ea diferă de soluţia generală 3 obţinută în exemplul de mai sus. Cele două soluţii generale generează aceeaşi mulţime de soluţii.

Dacă identificăm necunoscutele x_4 şi x_5 din cele două soluţii

$$x_4 = \alpha = b \quad \text{şi } x_5 = \beta = \frac{3}{2} - \frac{1}{2}\,a - \frac{1}{2}\,b,$$

adică, dacă folosim substituţia

$$\alpha = b \quad \text{şi } \beta = \frac{3}{2} - \frac{1}{2}\,a - \frac{1}{2}\,b,$$

în soluţia (3), regăsim soluţia (4):

$$\begin{cases} x_1 = 5 + b + 3 - a - b = 8 - a \\ x_2 = 3 - b - 3 + a + b = a \\ x_3 = -\frac{5}{5} - b - \frac{3}{2} + \frac{1}{2}a + \frac{1}{2}b = -\frac{19}{6} + \frac{1}{2}a - \frac{1}{2}b \\ x_4 = b \\ x_5 = \frac{3}{2} - \frac{1}{2}a - \frac{1}{2}b \end{cases},$$