UNIVERSIDADE ESTADUAL DO RIO GRANDE DO SUL

PROF. DR. ÉDER JULIO KINAST <eder-kinast@uergs.edu.br>

MÉTODOS NUMÉRICOS – APONTAMENTOS DE AULA

07 Resolução de Sistemas Lineares. Método de Gauss-Jordan

Versão 01 – 05/10/2020

Sistemas Lineares

Método de Gauss-Jordan

Sistemas Lineares

Sistemas lineares são um conjunto de m equações lineares de coeficientes constantes com n variáveis. Os métodos utilizados aqui serão aqueles que resolvem o caso em que $m = n$, ou seja, que o número de equações é igual ao número de variáveis. O formato tradicional de expressar um sistema linear genérico é:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \cdots + a_{1n} \cdot x_n = b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \cdots + a_{2n} \cdot x_n = b_2 \\ \qquad\qquad\qquad \ddots \\ a_{n1} \cdot x_1 + a_{n2} \cdot x_2 + \cdots + a_{nn} \cdot x_n = b_n \end{cases}$$

Além disso, os métodos utilizados aqui necessitam que $a_{ii} \neq 0$ (os coeficientes da diagonal principal não sejam nulos. Caso isto ocorra, deve-se promover mudanças entre as linhas.

Classificação

Mesmo no caso $m = n$, os sistemas lineares são classificados em:

- Sistema Possível e Determinado – quando o sistema possui solução e ela é única;
- Sistema Possível e Indeterminado – quando o sistema possui infinitas soluções (existe pelo menos uma combinação linear entre duas equações do sistema).

$$\begin{cases} 7 \cdot x - 5 \cdot y + 9 \cdot z = 50 \\ 2 \cdot x + 3 \cdot y - 4 \cdot z = 10 \\ 4 \cdot x + 6 \cdot y - 8 \cdot z = 20 \end{cases}$$

As equações L2 e L3 não são LI (linearmente independentes), pois L3 = 2 · L2

Classificação

- Sistema Impossível – quando o sistema não possui soluções (existe pelo menos uma combinação linear entre os coeficientes de duas equações do sistema, mas não do termo independente).

$$\begin{cases} 2 \cdot x + 3 \cdot y = 10 \\ 2 \cdot x + 3 \cdot y = 30 \end{cases}$$

As equações L1 e L2 não são coerentes.

Aqui serão tratados somente os sistemas possíveis e que possuem solução única.

Método de Gauss-Jordan

Este método consiste no escalonamento e eliminação de todos os coeficientes, exceto aqueles da diagonal principal, que devem valer 1.

No caso especial de sistemas 3×3, isto significa transformar o sistema da seguinte forma:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 = b_2 \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 = b_3 \end{cases} \rightarrow \begin{cases} 1 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = r_1 \\ 0 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3 = r_2 \\ 0 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 = r_3 \end{cases}$$

Na notação de matrizes completas $A\vec{x} = \vec{b}$ tem-se:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & r_1 \\ 0 & 1 & 0 & r_2 \\ 0 & 0 & 1 & r_3 \end{bmatrix}$$

Método de Gauss-Jordan

Basicamente, o método segue o seguinte algoritmo:

0. Caso algum elemento da diagonal principal seja zero, rearranjar as linhas;

1. Repetir os passos abaixo para cada linha:

   1.1. Normalizar a um o elemento da diagonal principal (e todo resto da linha);

   1.2. Zerar todos os elementos abaixo e acima do elemento recém normalizado;

2. Ao deixar a matriz com *escalonamento completo*, os elementos da última coluna (do vetor $\vec{b}$) são as soluções das variáveis (do vetor $\vec{x}$).

Gauss-Jordan 2×2

Exemplo (**MetNum07.xlsx!2x2**):

$$\begin{cases} 2 \cdot x + 4 \cdot y = 10 & (L1) \\ 2 \cdot x + 6 \cdot y = 2 & (L2) \end{cases}$$

Normalizar a_{11}: $L1 \leftarrow \frac{L1}{a_{11}}$

$$\begin{cases} 1 \cdot x + 2 \cdot y = 5 & (L1) \\ 2 \cdot x + 6 \cdot y = 2 & (L2) \end{cases}$$

Zerar a_{21}: $L2 \leftarrow L2 - a_{21} \cdot L1$

$$\begin{cases} 1 \cdot x + 2 \cdot y = 5 & (L1) \\ 0 \cdot x + 2 \cdot y = -8 & (L2) \end{cases}$$

Normalizar a_{22}: $L2 \leftarrow \frac{L2}{a_{22}}$

$$\begin{cases} 1 \cdot x + 2 \cdot y = 5 & (L1) \\ 0 \cdot x + 1 \cdot y = -4 & (L2) \end{cases}$$

Zerar a_{12}: $L1 \leftarrow L1 - a_{12} \cdot L2$

$$\begin{cases} 1 \cdot x + 0 \cdot y = 13 & (L1) \\ 0 \cdot x + 1 \cdot y = -4 & (L2) \end{cases}$$

	A	B	C
1	2	4	10
2	2	6	2
3			
4	1	2	5
5	0	2	-8
6			
7	1	0	13
8	0	1	-4

Solução:
$x = 13$ e $y = -4$

Gauss-Jordan 3×3

Exemplo (**MetNum07.xlsx!3x3**):

$$\begin{cases} 2 \cdot x + 4 \cdot y + 2 \cdot z = -8 \\ 3 \cdot x + 5 \cdot y + 2 \cdot z = -9 \\ 4 \cdot x \qquad\quad + 5 \cdot z = -1 \end{cases}$$

Normalizar a_{11}, então zerar a_{21} e a_{31}.

Normalizar a_{22}, então zerar a_{12} e a_{32}.

Normalizar a_{33}, então zerar a_{13} e a_{23}.

	A	B	C	D
1	2	4	2	-8
2	3	5	2	-9
3	4	0	5	-1
4				
5	1	2	1	-4
6	0	-1	-1	3
7	0	-8	1	15
8				
9	1	0	-1	2
10	0	1	1	-3
11	0	0	9	-9
12				
13	1	0	0	1
14	0	1	0	-2
15	0	0	1	-1

Gauss-Jordan 4×4

Exemplo (**MetNum07.xlsx!4x4**):

$$\begin{cases} 4\cdot x+2\cdot y+8\cdot z+5\cdot w=11 \\ 2\cdot x+4\cdot y+10\cdot z+4\cdot w=4 \\ -x+2\cdot y-2\cdot z-2\cdot w=-6 \\ x+2\cdot y+z=0 \end{cases}$$

	A	B	C	D	E
1	4	2	8	5	11
2	2	4	10	4	4
3	1	2	1	0	0
4	-1	2	-2	-2	-6
5					
6	1	0,5	2	1,25	2,75
7	0	3	6	1,5	-1,5
8	0	1,5	-1	-1,25	-2,75
9	0	2,5	0	-0,75	-3,25
10					
11	1	0	1	1	3
12	0	1	2	0,5	-0,5
13	0	0	-4	-2	-2
14	0	0	-5	-2	-2
15					
16	1	0	0	0,5	2,5
17	0	1	0	-0,5	-1,5
18	0	0	1	0,5	0,5
19	0	0	0	0,5	0,5
20					
21	1	0	0	0	2
22	0	1	0	0	-1
23	0	0	1	0	0
24	0	0	0	1	1

Rotina C/C++ para o Método de Gauss-Jordan

```
#include<iostream>
#include<math.h>

int main(){
    int ne=3; //número de equações e variáveis
    double A[20][20]={{2,4,2}, //Matriz dos coeficientes
                      {3,5,2},
                      {4,0,5}};
    double b[20]={-8,-9,-1}; //Matriz dos coeficientes
    double x[20],m,s;
    int i,j,k;
    for(k=0;k<=ne-2;k++){
        for(i=k+1;i<=ne-1;i++){
        m=A[i][k]/A[k][k];
        A[i][k]=0;
        for(j=k;j<=ne-1;j++) A[i][j]-=m*A[k][j];
        b[i]-=m*b[k];}}

    x[ne]=b[ne]/A[ne][ne];
    for(k=ne-1;k>=0;k--){
        s=0;
        for(j=k;j<=ne-1;j++) s+=A[k][j]*x[j];
        x[k]=(b[k]-s)/A[k][k];}

    for(i=0;i<=ne-1;i++) printf("x%d = %16.12lf\n",i,x[i]);

    system("PAUSE");
    return 0;}
```