Sistemas lineares e análise gráfica

Sistemas de Equações Lineares

Um problema básico de Álgebra Linear é a resolução de sistemas de equações lineares. A sua importância não se resume apenas na técnica de resolução, mas também aparece na redução de problemas matemáticos de caráter linear com um número finito de variáveis.

Considerando um sistema de m equações lineares a n incógnitas com coeficientes reais, podemos escrevê-la sob forma matricial como A x = b , e resolver o sistema significa discutir a existência de soluções e obter o conjunto solução quando for possível . Um dos métodos mais utilizados para isso é o Método da Eliminação de Gauss .

O Método da Eliminação de Gauss

O método consiste em transformar o sistema linear original para se obter um sistema linear equivalente com mesmo conjunto solução , usando o método do escalonamento . São efetuadas as seguintes operações sobre linhas da matriz que não alteram o conjunto solução dos sistemas :

(i) trocar as posições de duas equações ;

(i i) multiplicar uma equação por uma constante não nula ;

(i i i) adicionar um múltiplo não nulo de uma equação a uma outra equação ;

Efetuam-se as operações acima até obter a forma escalonada do sistema, A' x = b' , equivalente ao sistema original

Vamos aplicar o método num exemplo:

```
> with(linalg) :
> A := matrix ([[1, 2, 0], [3, 2,-5], [0, 0, 0], [0, 1, 0]]);
```

$$A := matrix\,[[1, 2, 0], [3, 2, -5], [0, 0, 0], [0, 1, 0]] \quad \textbf{(1)}$$

Funções do Maple V utilizadas neste método:

1) swaprow(A , i , j) : efetua troca de linhas.

A = matriz.

i e j = índices das linhas que serão trocadas de posição.

```
> swaprow(A, 2, 4);
```

(2)

$$\left[\begin{array}{ccc} matrix & 2\,matrix & 0 \\ 0 & matrix & 0 \\ 0 & 0 & 0 \\ 3\,matrix & 2\,matrix & -5\,matrix \end{array}\right] \quad \textbf{(2)}$$

2) addrow(A , [Maple Math] , [Maple Math] , multiplicador) : adiciona os elementos de uma linha multiplicada por 'm' não nulo

aos elementos de outra linha .

A = matriz.

> *addrow*$(A, 1, 2, -3)$;

$$\left[\begin{array}{ccc} matrix & 2\,matrix & 0 \\ 0 & -4\,matrix & -5\,matrix \\ 0 & 0 & 0 \\ 0 & matrix & 0 \end{array}\right] \quad \textbf{(3)}$$

O comando gausselim(A) permite obter diretamente uma forma escalonada de uma matriz.

3) gausselim(A) : obtém-se uma matriz escalonada equivalente à A.

> *gausselim*(A);

$$\left[\begin{array}{ccc} matrix & 2\,matrix & 0 \\ 0 & matrix & 0 \\ 0 & 0 & -5\,matrix \\ 0 & 0 & 0 \end{array}\right] \quad \textbf{(4)}$$

4) gaussjord(A) : obtém-se uma matriz escada linha-reduzida (l-reduzida) equivalente à A .

Obs. : A matriz escada l-reduzida equivalente à A é única. Porém, podem existir várias

matrizes escalonadas equivalentes à A (gausselim(A)) .

> *gaussjord*(A);

(5)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{(5)}$$

5) rank(A) : retorna o posto da matriz A.

> $rank(A);$

Error, (in linalg:-rank) modp1: invalid arguments to function ConvertIn

6) linsolve(A , b) : calcula as soluções do sistema dado por Ax = b .

> $b := vector([2, -1, 0, 5]);$

$$b := \begin{bmatrix} 2 & -1 & 0 & 5 \end{bmatrix} \quad \mathbf{(6)}$$

> $linsolve(A, b);$

$$\begin{bmatrix} -\frac{8}{matrix} & \frac{5}{matrix} & -\frac{13}{5\,matrix} \end{bmatrix} \quad \mathbf{(7)}$$

Quando o sistema não possui solução, o comando acima retorna vazio.

\>

7) linsolve(A , b ,'r' , v) : caso o sistema seja Possível e Indeterminado, as soluções serão da das em função

de uma variável especificada no 4º parâmetro . O 3º parâmetro da função linsolve ( ) indicará o posto da matriz A.

Exemplo de sistema possível e indeterminado:

x + y + z = 2

x + 2y + 3z = 1

y + 2z = -1

> $B := matrix([[1, 1, 1], [1, 2, 3], [0, 1, 2]]);$

(8)

$$B := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 0 & 1 & 2 \end{bmatrix} \tag{8}$$

> $c := matrix([[2], [1], [-1]]);$

$$c := \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} \tag{9}$$

O comando "concat" permite considerar a matriz completa do sistema:

> $B\_1 := concat(B, c);$

$$B\_1 := \begin{bmatrix} 1 & 1 & 1 & 2 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \end{bmatrix} \tag{10}$$

> $rank(B);$

$$2 \tag{11}$$

> $rank(B\_1);$

$$2 \tag{12}$$

Temos compatibilidade no sistema, com grau de liberdade 3(número de variáveis) - 2(posto do sistema) = 1 (uma variável independente).

> $linsolve(B, c,'r', t);$

$$\begin{bmatrix} 3 + (t_1)_1 \\ -1 - 2\,(t_1)_1 \\ (t_1)_1 \end{bmatrix} \tag{13}$$

Exemplos de Análise Gráfica de Soluções de Sistemas Lineares

Podemos ter uma idéia gráfica do com portamento de Sistemas Lineares com n equações e 3 incógnitas (Espaço R^3) e de Sistemas Li neares com n equações e 2 incógnitas (Espaço R^2) .

Nesta exposição não vamos esgotar todos os casos possíveis, mas apenas ilustrar alguns.

Sistema Possível e Determinado

Considerando o Espaço R^3

Obs: Para facilitar, plotamos o plano ax + by + cz + d = 0, escrevendo uma das variáveis em função das outras duas e utilizando o formato de gráfico de função.

Exemplo:

Considere o sistema abaixo :

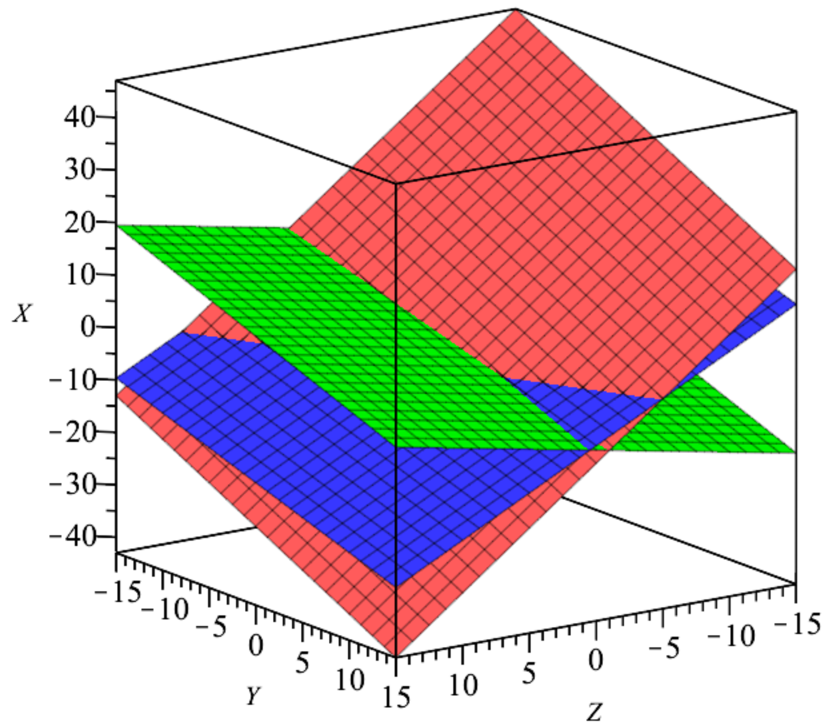
3x + 2y + 4z = 1

1x + 1y + 2z = 2

4x + 3y -2z = 3

A solução do sistema é a intersecção dos planos representados pelas equações que constituem o sistema acima: (x , y , z) = (-3 , 5 , 0) .

```
> with(plots) :
  plano1 := plot3d( (1-2*y-4*z) /3, z=-15 ..15, y=-15 ..15, color=blue, style=PATCH ) :
  plano2 := plot3d(2-y-2*z, z=-15 ..15, y=-15 ..15, color=red, style=PATCH ) :
  plano3 := plot3d( (3-3*y + 2*z) /4, z=-15 ..15, y=-15 ..15, color=green, style=PATCH ) :
  display3d(plano1, plano2, plano3, axes=boxed, labels=[Z, Y, X] );
```



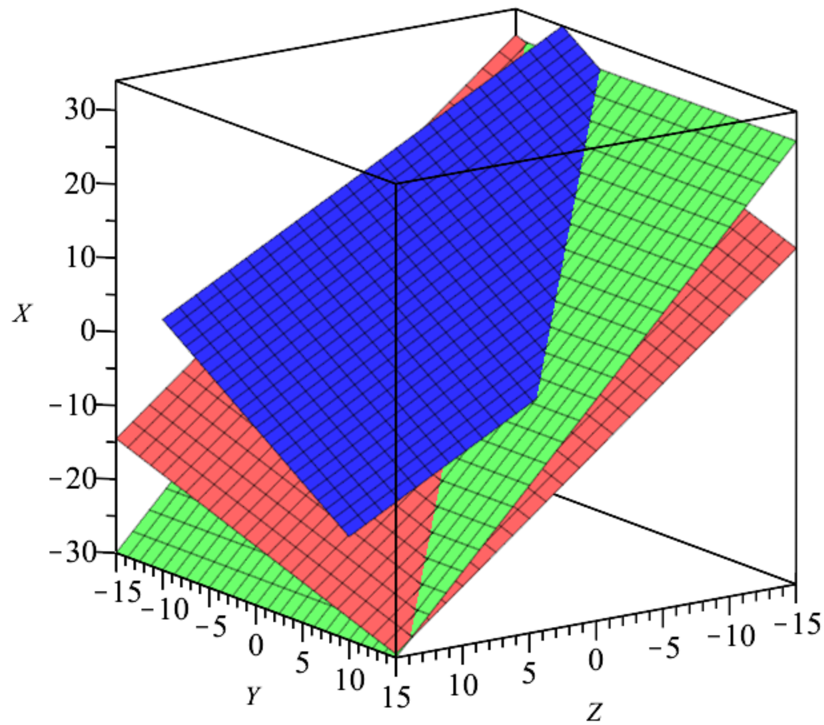
Sistema Impossível

Considerando Espaço R³

Exemplo : Três planos no espaço .

Solução: não existe intersecção dos planos

```
> plano1 := plot3d(9-x-z, z=-15..15, x=-10..10, color=blue, style=PATCH) :
  plano2 := plot3d((1-3*z-x)/2, z=-15..15, x=-15..15, color=red, style=PATCH) :
  plano3 := plot3d(-2*z, z=-15..15, x=-15..15, color=green, style=PATCH) :
  display3d(plano1, plano2, plano3, axes=boxed, labels=[Z, Y, X]);
```



Considerando o Espaço R²

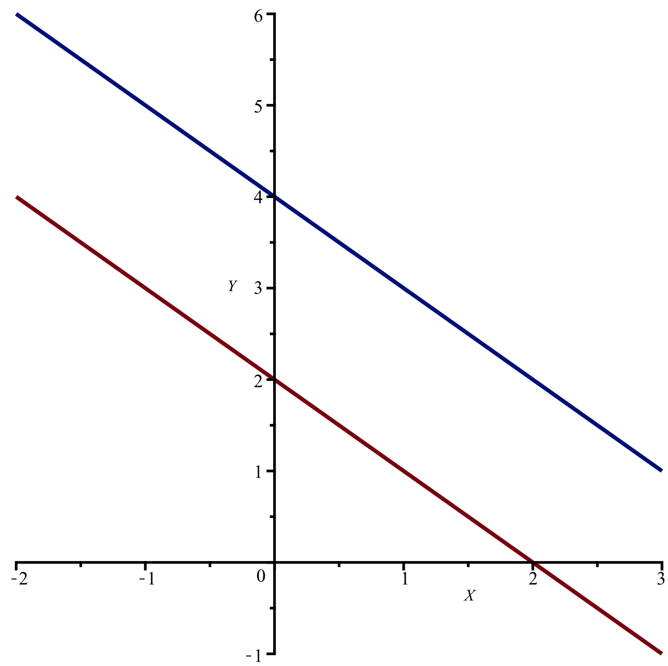
Exemplo: Duas retas paralelas.

x + y = 4

x + y = 2

Solução : Não existe intersecção das retas, pois elas são paralelas. Graficamente :

```
> x := 'x':
  reta1 := 4 - x :
  reta2 := 2 - x :
  plot( {reta1, reta2}, x =-2 ..3,-1 ..6, labels = [X, Y] );
```


6
5
4
3
2
1
0
-1
Y
-2
-1
1
2
3
X