Universidad de Jaén
Departamento de Matemáticas
(Área de Álgebra)

Sistemas de ecuaciones lineales: Método de Gauss-Jordan

La resolución de sistemas de ecuaciones es uno de los problemas que con más frecuencia se encuentra en los distintos campos de la Ciencia. En particular, nosotros como paso previo a resolver muchos problemas, tendremos que resolver sistemas de ecuaciones. En primer lugar veremos los comandos que Mathematica incorpora para discutir y resolver sistemas de ecuaciones, y posteriormente, otras alternativas a este estudio, así veremos cómo es posible mediante pequeños programas resolver sistemas triangulares, transformar un sistema en otro triangular superior o escalonado (método de Gauss) y transformar un sistema en otro diagonal o escalonado reducido (método de Gauss-Jordan) o usar las fórmulas de Cramer.

1. RESOLUCIÓN DE SISTEMAS DE ECUACIONES CON MATHEMATICA.

La resolución de sistemas de ecuaciones lineales en Mathematica puede hacerse de dos maneras diferentes:

(a) Mediante el comando LinearSolve, cuya sintaxis es:

LinearSolve[A, b**]**

donde A es la matriz de coeficientes y b el vector de términos independientes y nos devolverá un vector que cumple la ecuación matricial Ax = b.

Destacar que en sistemas compatibles indeterminados, este comando sólo calcula una de sus soluciones. Para reconocer si un sistema es determinado o no podemos usar el comando NullSpace que determina si el correspondiente sistema homogéneo tiene solución no nula:

NullSpace[A**]**

donde A es una matriz, nos devuelve una base del las soluciones del sistema A.x=0.

Ejemplo 4.1. Resolver el sistema de ecuaciones:

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 1 & 0 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix}$$

In[]: = **a={{1,2,3},{2,3,4},{1,0,3}};**

b={1,0,5};
LinearSolve[a, b]

Out[]: = {-1,-2,2}

Veamos ahora si la anterior es la única solución del sistema, es decir, si el sistema es compatible determinado:

In[]:= **NullSpace[a]**

Out[]:= {}

Por tanto, la solución que nos daba LinearSolve es única pues el sistema homogéneo Ax = 0 tiene como solución el espacio vectorial {0}. Como el sistema es compatible y determinado, otra forma de resolver dicho sistema es invirtiendo la matriz de coeficientes como se muestra en el siguiente ejemplo:

In[]:= **Inverse[a].b**

Out[]:= {-1,-2,2}

■

Sin embargo en el siguiente ejemplo el sistema es compatible indeterminado y la orden LinearSolve sigue dándonos solo una solución del sistema:

Ejemplo 4.2. Resolver el sistema de ecuaciones:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 2 \\ 1 & 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}$$

In[]:= **LinearSolve[{{1,1,0},{0,-1,2},{1,0,2}},{1,-1,0}]**

Out[]:= {0,1,0}

In[]:= **NullSpace[{{1,1,0},{0,-1,2},{1,0,2}}]**

Out[]:= {{-2,2,1}}

Por tanto, la solución del sistema será:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}$$

■

b) Escribiendo las ecuaciones y utilizando para resolverlas el comando Solve.

Mathematica entiende por ecuación una expresión lógica de igualdad,

expresión1==expresión2

y un sistema de ecuaciones por una lista de ecuaciones de la forma:

{expresión1==expresión2, expresión3==expresión4,...}

La sintaxis del comando Solve es:

Solve[{ecuación1,ecuación2,...},{var1,var2,...}**]**

O bien de forma matricial:

Solve[A.{var1,var2,...}=={b1,b2,...},{var1,var2,...}**]**

Otros comandos que incorpora Mathematica para resolver ecuaciones son los comandos Reduce y NSolve con sintaxis:

Reduce[{ecuación1,ecuación2,...},{var1,var2,...}**]**

Nsolve[{ecuación1,ecuación2,...},{var1,var2,...}**]**

El primero reduce el sistema de ecuaciones a otro más sencillo equivalente al primero, y el segundo resuelve numéricamente el sistema de ecuaciones dando una solución aproximada.

Ejemplo 4.3. Usar lo anterior para resolver el sistema de ecuaciones:

```
x + y      = 1
  - y + 2z = -1
 x     + 2z = 0
```

In[]:= **Solve[{x+y==1,-y+2z==-1,x+2z==0},{x,y,z}]**

Out[]:= Solve::svars:
Equations may not give solutions for all "solve" variables.

{{x->-2 z, y->1+2 z}}

Observar que Mathematica advierte que las ecuaciones pueden no dar solución a todas las variables como ocurre aquí con la variable z, que sería la que jugaría el papel de parámetro en la solución del sistema compatible indeterminado.

In[]:= **Reduce[{x+y==1,-y+2z==-1,x+2z==0},{x,y,z}]**

Out[]:= x==-2 z&&y==1+2 z

In[]:= **NSolve[{x+y==1,-y+2z==-1,x+2z==0},{x,y,z}]**

Out[]:= {x->0. – 2. z, y->1. + 2. z}

■

Además con el comando Reduce Mathematica realiza un estudio de casos en función de los parámetros que aparezcan en el sistema:

Ejemplo 4.4. Usar Reduce para discutir el siguiente sistema con parámetros:

$$\begin{aligned} px+y &= 0 \\ x - y &= 0 \end{aligned}$$

In[]:= **Reduce[{p*x+y==0,x-y==0},{x,y}]**

Out[]:= **y**==x&&p==-1 || 1+p≠ 0&&y==0 && x==0

Esto es, si p = –1, el sistema es compatible indeterminado con solución (λ,λ) y si p ≠ –1 el sistema es compatible determinado con solución (0, 0).

■

Observar por último que los Solve, Reduce y NSolve resuelven también sistemas de ecuaciones no lineales.

2. RESOLUCIÓN DE SISTEMAS USANDO LA FORMA NORMAL DE HERMITE DE LA MATRIZ AMPLIADA.

El siguiente resultado visto en clase nos proporciona otro método para resolver sistemas de ecuaciones lineales

Proposición. Dado un sistema de ecuaciones lineales con (A|B) como matriz ampliada. Si H es la forma normal de Hermite por filas de (A|B), entonces el sistema cuya matriz ampliada es H es un sistema escalonado reducido equivalente al de partida.

Ejemplo 4.5. Aplicar la proposición anterior para obtener un sistema escalonado reducido equivalente al sistema:

$$\begin{aligned} x + 2y + 3z &= 1 \\ 2x + 3y + 4z &= 0 \\ x \qquad\; + 3z &= 5 \end{aligned}$$

In[]:= **ampliada={{1,2,3,1},{2,3,4,0},{1,0,3,5}};**
MatrixForm[%]

Out[]:= $\begin{pmatrix} 1 & 2 & 3 & 1 \\ 2 & 3 & 4 & 0 \\ 1 & 0 & 3 & 5 \end{pmatrix}$

In[]:= **RowReduce[ampliada]//MatrixForm**

Out[]:= $$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Por tanto, el sistema escalonado reducido equivalente al dado y por tanto la solución del sistema dado es:

$$\begin{aligned} x \quad\quad &= -1 \\ y \quad &= -2 \\ z &= 2 \end{aligned}$$

■

3. RESOLUCIÓN DE SISTEMAS TRIANGULARES.

Vamos a resolver un sistema de la forma A.x = b, donde A es una matriz regular y triangular:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

En primer lugar, se introduce la matriz A y el vector b:

A={{a_{11}, a_{12},..., a_{1n}},{0, a_{22},..., a_{2n}},...,{0,0,..., a_{nn}}};
b={ b_1, b_2,..., b_n};

Llamamos **n** al orden de la matriz cuadrada del sistema:

n=Dimensions[A][[1]];

y definimos el vector **x**, que al final contendrá la solución y que inicializamos como el vector cero:

x=Table[0,{i,1,n}];

Utilizando el siguiente bucle hallamos la solución mediante sustitución sucesiva:

```
For[i=n, i>=1, i--,
    x[[i]]=(b[[i]]-Sum[A[[i,k]]*x[[k]],{k,i+1,n}]) / A[[i,i]]
    ];
```

La solución es

x

Ejemplo 4.6. Resolver el sistema de ecuaciones triangular superior dado por:

$$\begin{pmatrix} 1 & -2 & 3 & 4 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 3 \\ 10 \end{pmatrix}$$

In[]: =

```
A={{1,-2,3,4},{0,2,0,3},{0,0,1,-2},{0,0,0,2}};
b = {4,1,3,10};
n=Dimensions[A][[1]];
x=Table[0,{i,1,n,1}];
For[i=n, i>=1, i--,
        x[[i]]=(b[[i]]-Sum[A[[i,k]]*x[[k]],{k,i+1,n}]) / A[[i,i]]
    ];
Print["La solución del sistema es ", x]
```

Out[]: = La solución del sistema es {-69,-7,13, 5}

■

4. TRANSFORMAR UN SISTEMA EN OTRO TRIANGULAR SUPERIOR: MÉTODO DE GAUSS.

Ahora queremos reducir un sistema a otro equivalente y triangular superior mediante el procedimiento de eliminación gaussiana. Sea A.x = b un sistema de Cramer (es decir, A es cuadrada y regular):

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Introducimos la matriz A y el vector b:

A={{a_{11}, a_{12},..., a_{1n}},{ a_{21}, a_{22},..., a_{2n} },...,{ a_{n1}, a_{n2},..., a_{nn} }};
b={b_1, b_2,..., b_n};

Hallamos el orden de la matriz A y definimos el vector cero de dimensión n:

```
n=Dimensions[A][[1]];
x=Table[0,{i,n}];
```

El bucle que nos proporciona la obtención de la matriz escalonada será:

```
For[i=1,i≤n,i++,
  For[j=i,j≤n,j++,
        If[A[[i]]!=x,
              If[A[[i,j]]==0 ,
                    For[k=i,k≤n,k++,
```

```
If[A[[k,j]]≠0,z=A[[i]];A[[i]]=A[[k]];A[[k]]=z;Break[]]
                ];s=A[[i,j]];If[s≠0,A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s],
              s=A[[i,j]];A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s
       ],For[k=n,k>=i+1,k--,If[A[[k]]≠x,A[[i]]=A[[k]];A[[k]]=x;Break[]]]
     ];
 Do[p=A[[l,j]];If[p≠0, A[[l]]=A[[l]]-p*A[[i]];b[[l]]=b[[l]]-p*b[[i]]],{l,i+1,n,1}
   ];If[A[[i,j]]≠0,Break[]]
     ]
        ]
```

La matriz triangular se sigue llamando A y el vector de términos independientes se sigue llamando b. Ahora solo quedará resolver el problema usando el bucle de la sección anterior.

Ejemplo 4.7. Resolver el sistema de ecuaciones

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 6 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -8 \\ 0 \\ 1 \end{pmatrix}$$

En primer lugar reducimos el sistema a un sistema escalonado superior:

In[]:=

```
A={{1,2,3},{2,3,4},{3,4,6}};
b={-8,0,1};
n=Dimensions[A][[1]];
x=Table[0,{i,n}];
For[i=1,i≤n,i++,
   For[j=i,j≤n,j++,
     If[A[[i]]!=x,
         If[A[[i,j]]==0 ,
For[k=i,k≤n,k++,If[A[[k,j]]≠0,z=A[[i]];A[[i]]=A[[k]];
      A[[k]]=z;Break[]]
      ];s=A[[i,j]];If[s≠0,A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s],
      s=A[[i,j]];A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s],
For[k=n,k>=i+1,k-- ,If[A[[k]]≠x,A[[i]]=A[[k]];
A[[k]]=x;Break[]]]]];
  Do[p=A[[l,j]];If[p≠0, A[[l]]=A[[l]]-p*A[[i]];b[[l]]=b[[l]]-
p*b[[i]]],{l,i+1,n,1}];If[A[[i,j]]≠0,Break[]]
]
]
MatrixForm[A]
MatrixForm[b]
```

Out[]:=

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -8 \\ -16 \\ -7 \end{pmatrix}$$

Ahora resolvemos usando la siguiente rutina que completa la vista en el primer apartado para sistemas compatibles y determinados:

In[]:=
```
Incompatible=False;
For[i=1,i<=n,i++,
 If[A[[i]]==x && b[[i]]≠0,Incompatible=True]]
 If[Incompatible,Print["El sistema es incompatible"],
 If[n==MatrixRank[A],For[i=n,i>=1,i--,x[[i]]=(b[[i]]-
Sum[A[[i,k]]*x[[k]],{k,i+1,n}])/A[[i,i]]];
Print["El sistema es compatible y determinado"];Print["La
solución del  sistema es ", x],
 Print["El sistema es compatible e indeterminado"];
 Reduce[A.{x1,x2,x3}==b,{x1,x2,x3}]]]
```

Out[]:= El sistema es compatible y determinado

La solución del sistema es {17,-2,-7}

■

5. MÉTODO DE GAUSS-JORDAN.

Ahora al triangular por Gauss vamos a hacer ceros encima de la diagonal al mismo tiempo que lo hacemos ceros debajo de ésta y una vez hechos todos los ceros, si el sistema es compatible determinado, tendremos resuelto el sistema, pues cada incógnita corresponderá a un valor del vector b. Esto es lo que se conoce con el nombre de método de Gauss-Jordan.

Introducimos de nuevo la matriz A y el vector b:

A={{a_{11}, a_{12},..., a_{1n}},{ a_{21}, a_{22},..., a_{2n} },...,{ a_{n1}, a_{n2},..., a_{nn} }};
b={b_1, b_2,..., b_n};

Hallamos el orden de la matriz cuadrada A:

```
n=Dimensions[A][[1]];
```

El bucle que hace ceros al mismo tiempo encima y debajo de la diagonal viene dado por:

```
For[i=1,i≤n,i++,
  For[j=i,j≤n,j++,
```

```
If[A[[i]]!=x,
        If[A[[i,j]]==0 ,
                For[k=i+1,k≤n,k++,
        If[A[[k,j]]≠0,z=A[[i]];A[[i]]=A[[k]];A[[k]]=z;Break[]]
                ];s=A[[i,j]];If[s≠0,A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s],
                s=A[[i,j]];A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s
        ],For[k=n,k>=i+1,k--
,If[A[[k]]≠x,A[[i]]=A[[k]];A[[k]]=x;Break[]]]
];
Do[p=A[[l,j]];If[p≠0, A[[l]]=A[[l]]-p*A[[i]];b[[l]]=b[[l]]-
p*b[[i]]],{l,i+1,n,1}
];Do[p=A[[l,j]];If[p≠0, A[[l]]=A[[l]]-p*A[[i]];b[[l]]=b[[l]]-
p*b[[i]]],{l,1,i-1,1}
];If[A[[i,j]]≠0,Break[]]
    ]
      ]
```

Cuando el sistema sea compatible determinado, la matriz que se obtiene es la matriz identidad que sigue llamándose A y el vector de términos independientes se sigue llamando b. Ahora solo quedará resolver el nuevo sistema equivalente con el primero lo que conseguimos solo igualando la variable correspondiente con la correspondiente coordenada del vector resultante b:

```
For[i=1,i<=n,i++,x[[i]]=b[[i]]]

Print[“La solución del sistema es ”, x]
```

Ejemplo 4.8. Aplicar el método de Gauss-Jordan para resolver el sistema del ejemplo anterior.

```
In[]:=    A = {{1,2,3},{2,3,4},{3,4,6}};
          b ={-8,0,1};
          n =Dimensions[A][[1]];
          x=Table[0,{i,1,n,1}];
            For[i=1,i≤n,i++,
           For[j=i,j≤n,j++,
          If[A[[i]]!=x,
                  If[A[[i,j]]==0 ,
                          For[k=i+1,k≤n,k++,
                  If[A[[k,j]]≠0,z=A[[i]];A[[i]]=A[[k]];A[[k]]=z;Break[]]
                          ];s=A[[i,j]];If[s≠0,A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s],
                          s=A[[i,j]];A[[i]]=A[[i]]/s;b[[i]]=b[[i]]/s
                  ],For[k=n,k>=i+1,k--,If[A[[k]]≠x,A[[i]]=A[[k]];
                  A[[k]]=x;Break[]]]
                  ];
          Do[p=A[[l,j]];If[p≠0, A[[l]]=A[[l]]-p*A[[i]];
            b[[l]]=b[[l]]-p*b[[i]]],{l,i+1,n,1}];Do[p=A[[l,j]];If[p≠0,
            A[[l]]=A[[l]]-p*A[[i]];b[[l]]=b[[l]]-p*b[[i]]],{l,1,i-1,1}
          ];If[A[[i,j]]≠0,Break[]]
```

```
        ]
            ]
MatrixForm[A]
MatrixForm[b]
```

Out[]:=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} 17 \\ -2 \\ -7 \end{pmatrix}$$

Ahora resolvemos usando la rutina del apartado anterior, aunque como hemos visto antes para sistemas compatibles determinado no sería necesario su uso:

In[]:=

```
Incompatible=False;
For[i=1,i<=n,i++,
      If[A[[i]]==x && b[[i]]≠0,Incompatible=True]]
    If[Incompatible,Print["El sistema es incompatible"],
    If[n==MatrixRank[A],For[i=n,i>=1,i--,x[[i]]=(b[[i]]-
Sum[A[[i,k]]*x[[k]],{k,i+1,n}])/A[[i,i]]];
Print["El sistema es compatible y determinado"];Print["La
solución del  sistema es ", x],
  Print["El sistema es compatible e indeterminado"];
  Reduce[A.{x1,x2,x3}==b,{x1,x2,x3}]]]
```

Out[]:= El sistema es compatible y determinado

La solución del sistema es {17,-2,-7}

■

6. REGLA DE CRAMER.

Otra aplicación de los determinantes es la resolución de sistemas de ecuaciones aplicando la regla de Cramer.

Dado un sistema de ecuaciones A.X = B donde $A = (a_{ij})_{i,j}$ es la matriz de coeficientes, X y B son las matrices columna de incógnitas y de términos independientes, respectivamente:

$$X = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \qquad B = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

diremos que es un **sistema de Cramer** si A es cuadrada y regular.

<u>Teorema</u> (Regla de Cramer) Dado un sistema de Cramer:

$$\begin{cases} a_{11}x_1+\ldots+a_{1n}x_n = b_1 \\ a_{21}x_1+\ldots+a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1+\ldots+a_{nn}x_n = b_n \end{cases}$$

la solución (única) del sistema viene dada por:

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & \cdots & a_{1n} \\ b_2 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_n & a_{n2} & \cdots & a_{nn} \end{vmatrix}}{|A|};\ x_2 = \frac{\begin{vmatrix} a_{11} & b_1 & \cdots & a_{1n} \\ a_{21} & b_2 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & b_n & \cdots & a_{nn} \end{vmatrix}}{|A|};\ldots;\ x_n = \frac{\begin{vmatrix} a_{11} & a_{12} & \cdots & b_1 \\ a_{21} & a_{22} & \cdots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & b_n \end{vmatrix}}{|A|}$$

Ejemplo 4.9. Resolver el sistema de Cramer:

$$\begin{cases} x + 2y - z = 1 \\ -2y + z = 2 \\ 2x + y - z = 3 \end{cases}$$

Introducimos la matriz de coeficientes A y el vector de términos independientes:

In[]:= **A={{1,2,-1},{0,-2,1},{2,1,-1}};**
b={1,2,3};

El siguiente paso sería calcular la matriz que se obtiene cambiando la primera columna por el vector de términos independientes y calcular la primera coordenada del vector solución, esto lo hacemos del siguiente modo:

In[]:= **A1=Transpose[A];**
A1[[1]]=b;
x=Table[0,{i,Dimensions[A][[1]]}];
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= 3

De igual forma lo haríamos para la segunda y tercera coordenada:

In[]:= **A1=Transpose[A];**
A1[[2]]=b;
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= 1

In[]:= **A1=Transpose[A];**
A1[[3]]=b;
x[[1]]=Det[Transpose[A1]]/Det[A]

Out[]= 4

Vamos ahora a hacerlo todo de una vez usando para ello la orden Module con la que crearemos un nuevo comando **cramer** que nos resolverá los sistemas de Cramer:

In[]:=
```
cramer[A_,b_]:=Module[{A1,x,n},
A1=Transpose[A];
n=Dimensions[A][[1]];
x=Table[0,{i,1,n}];
Do[A1[[i]]=b;
x[[i]] =Det[Transpose[A1]]/Det[A];A1=Transpose[A],{i,1,n}];
Print["La solución del sistema de Cramer es ", x] ]
```

In[]:= **cramer[A,b]**

Out[]= La solución del sistema es {3,1,4}

■