CURSO 2007-2008

INGENIERÍA TÉCNICA INDUSTRIAL - ESP. MECÁNICA

FUNDAMENTOS MATEMÁTICOS DE LA INGENIERÍA

Práctica nº 4: Álgebra vectorial y matricial.

Resolución de sistemas de ecuaciones. Autovalores y autovectores

RESUMEN: *Abordamos en esta práctica el cálculo vectorial y matricial y la resolución de sistemas de ecuaciones lineales. Estudiaremos métodos de resolución basados en el teorema de Rouché-Frobenius y en los métodos directos de MAPLE. Asimismo consideramos métodos directos de MAPLE para el cálculo de autovalores y autovectores.*

COMANDOS MÁS IMPORTANTES: *vector, matrix, evalm, det, rank, eigenvalues, eigenvectors.*

Álgebra vectorial

Casi todas las funciones de Álgebra lineal están en una librería que se llama **linalg** (linear algebra). Si se intenta utilizar algúna función de esta librería sin cargarla previamente, Maple se limita a repetir el nombre de la función sin realizar ningún cálculo. Para cargar dicha librería se teclea el comando siguiente:

```
> with(linalg):
```

Comenzaremos por familiarizarnos con los vectores. Para definir un vector, Maple incorpora el comando **vector**. Su sintaxis es como sigue:

```
> v:=vector([1,2,1,-3]);
> w:=vector([0,2,-1,4]);
```

Para visualizar alguna componente particular de un vector, escribiremos su nombre y entre corchetes la componente que deseamos visualizar. Por ejemplo,

```
> v[1];# muestra la primera componente de v
  w[3]; # muestra la tercera componente de w
```

A continuación calculamos 4w-2v.

```
> evalm(4*w-2*v);
```

Álgebra matricial

Como ya sabemos, una matriz de orden *m x n* es un conjunto de elementos distribuidos entre *m* filas y *n* columnas

$$A=(a_{i,j})=\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Una forma de introducir una matriz de *m x n* elementos se hace mediante la sentencia:

A:=matrix([[$a_{1,1}$, $a_{1,2}$, ..., $a_{1,n}$ **],[** $a_{2,1}$, ..., $a_{2,n}$ **],..,[** $a_{m,1}$, ..., $a_{m,n}$ **]])**

Para referirnos a la entrada *(i,j)* de una matriz **A** lo hacemos con **A[*i,j*]**.

```
> A:=matrix([[1,2,3],[5,6,7],[1,6,3]]);
```

El elemento $a_{2,2}$ de la matriz anterior es:

```
> A[2,2];
```

Una submatriz de una matriz *A* dada se obtiene suprimiendo algunas filas y/o columnas de ella. Con MAPLE se pueden extraer submatrices con el comando:

submatrix(*A*, f_1 .. f_m, c_1 .. c_n);

cuando especificamos las filas comprendidas entre la f_1 y la f_m y las columnas entre la c_1 y la c_n, o sustituyendo alguno de estos rangos por listas de filas o columnas no necesariamente consecutivas:

submatrix(A,[f_1,f_2,...,f_m],[c_1, c_2, ..., c_n]);

Dada la matriz:

```
> A:=matrix([[1,5,-3,2,6],[1,4,0,0,-5],[-2,7,8,9,-3],[3,-1,-1,0
  ,-4]]);
```

La submatriz formada por las tres últimas filas y las tres primeras columnas es:

```
> submatrix(A, 2..4, 1..3);
```

Y la submatriz formada por las dos filas centrales y las columnas 1, 3 y 5 es:

```
> submatrix(A,2..3,[1,3,5]);
```

El menor complementario del elemento $a_{i,j}$ de la matriz *A* es la submatriz obtenida al suprimir la fila *i* y la columna *j* de *A*:

minor(A,i,j);

```
> minor(A,2,3);
```

El conjunto de matrices de orden m x n define un espacio vectorial de dimensión m·n, en el que está definida la suma y el producto por un escalar.Para ellos utilizamos los operadores clásicos de suma (+) y producto (*) teniendo en cuenta que sólo se pueden sumar matrices de igual dimensión. Podemos realizar cualquier combinación lineal de matrices y lo visualizamos con la orden evalm.

Para multiplicar matrices, el número de columnas de la primera ha de coincidir con el número de filas de la segunda. Dadas dos matrices A y B, su producto se calcula por:

A&*B

Ejercicio: Sean $A=\begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 1 & 6 & 3 \end{bmatrix}$, $B=\begin{bmatrix} 1 & -1 & 0 \\ 3 & 2 & 7 \\ 1 & 1 & 4 \end{bmatrix}$, $C=\begin{bmatrix} -1 & 2 & 0 \\ 2 & 3 & 8 \\ 1 & 6 & 2 \end{bmatrix}$ e I la matriz identidad apropiada. Calcular **i)** A+C, 7B, 3A-5I, **ii)** (3A-5I)B, AC, CA

Resolución:

```
> restart:
```

```
> with(linalg):
> A:=matrix([[1,2,3],[5,6,7],[1,6,3]]);
> B:= matrix([[1, -1, 0], [3, 2, 7], [1, 1, 4]]);
> C:=matrix([[-1, 2, 0], [2, 3, 8], [1, 6, 2]]);
> evalm(A+C);
> evalm(7*B);
> evalm(3*A-5*diag(1,1,1));
```

¿Cómo interpreta MAPLE la operación A+2?

Otras operaciones sobre matrices son:

transpose(A)	calcula la matriz traspuesta de A.
det(A)	calcula el determinante de la matriz cuadrada A.
evalm(A^(-1)) o **inverse(A)**	calcula la matriz inversa de la matriz regular A.

Recuérdese que una matriz se dice regular si tiene determinante no nulo.

Ejercicio: Calcular el determinante y las inversas de A,C, AC ,CA. Comprobar con A que su determinate coincide con el determinante de su traspuesta ¿Ocurre ésto para cualquier matriz?.

Resolución:

```
> det(A);
> det(C);
> det(A&*C);
> det(C&*A);
> inverse(A);
> inverse(C);
> inverse(A&*C);
> inverse(C&*A);
> det(transpose(A));
```

Otra característica de una matriz es su rango, que es el mayor número de filas o de columnas linealmente independientes, y se calcula con

rank(matriz);

Obsérvese que una matriz regular tiene rango máximo. De nuevo el rango de una matriz coincide con el de su traspuesta.

Ejercicio: Calcular el rango de A, B y C y comprobar con B que el rango coincide con el de su traspuesta.

Resolución:

```
> rank(A);
> rank(B);
> rank(C);
> transpose(B);
> rank(transpose(B));
```

Sistema de ecuaciones lineales.

Un sistema lineal de m ecuaciones y n incógnitas viene dado por

$$a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1$$

$$a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2$$
$$\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots$$
$$a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n = b_m$$

donde los $a_{i,j}$ son los coeficientes conocidos, x_j son las incógnitas y los b_j son los términos independientes. Podemos expresar el sistema en forma matricial A X = b

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix}$$

La matriz ampliada, matriz que se construye añadiendo a A la columna del vector b de términos independientes, se calcula con la orden

`AM:= concat(A,b)`

$$AM = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} & b_m \end{bmatrix}$$

Si A es una matriz no singular entonces podemos calcular su inversa $A^{(-1)}$y resolvemos el sistema de la forma

evalm(X)=evalm($A^{(-1)}$ &*b)

Si disponemos del sistema también pueden obtenerse la matriz de coeficientes *A* y la ampliada *AM*, respectivamente, mediante los comandos:

genmatrix([*ecu 1,ecu 2,...,ecu m***], [***incog 1,incog 2,..., incog n***]);**

genmatrix([*ecu 1,ecu 2,...,ecu m***], [***incog 1,incog 2,..., incog n***],flag);**

Teorema de Rouché-Frobenius

El estudio del sistema AX=b se basa en la relación que hay entre el número de ecuaciones y de incognitas.

El teorema de Rouché-Frobenius nos dice:

1.- rank(A) < rank(AM) (sistema incompatible) : No hay solución.

2.- **rank(A) = rank(AM) = r**

2.1.- **r = nº de incógnitas** (sistema compatible determinado)

La matriz A es no singular, luego existe $A^{(-1)}$y la solución del sistema es única y viene dada por **X** =$A^{(-1)}$**b.**

2.2.- **r < nº de incógnitas** (sistema compatible indeterminado)

La matriz A es singular, luego no admite inversa. Del sistema tenemos que extraer una submatriz cuadrada de rango **r** (no singular entonces) pasando incógnitas al segundo miembro. Nos situamos entonces en el caso 2.1.

Ejercicio: Resolver

$$2x + y - z = 1$$
$$x + 2y + z = 5$$
$$3x + y + 2z = -1$$

Resolución:

```
> ecuaciones:=2*x+y-z=1,x+2*y+z=5,3*x+y+2*z=-1;
> A:= genmatrix([ecuaciones],[x,y,z]);
> X:=matrix([[x],[y],[z]]);# Lo definimos así para obtener
  un vector columna
> b:=matrix([[1],[5],[-1]]);
> evalm(A)*evalm(X)=evalm(b);# Método para expresar en forma
  matricial el sistema
> AM:=concat(A,b);# Construcción de la matriz ampliada.
> AM:=genmatrix([ecuaciones],[x,y,z],flag); # Otra forma de
  construir la matriz ampliada.
> [rank(A),rank(AM)];# Rangos de A y AM para aplicar.
  Rouché-Frobenius
```

Se trata de un sistema compatible determinado con solución única.

```
> evalm(X)=evalm(inverse(A)&*b);
```

Ejercicio: Resolver

$$x + 2y + 3z + 3t = 2$$
$$x - y + z = 0$$
$$x + 3y - z + 4t = -2$$
$$3x + 4y + 3z + 7t = 0$$

Resolución:

Podemos construir las matrices directamente:

```
> A:=matrix( [[1,2,3,3],[1,-1,1,0],[1,3,-1,4],[3,4,3,7]]);
> b:=matrix([[2],[0],[-2],[0]]);
> AM:=concat(A,b);
```

O directamente a partir del sistema

```
> sistema:=x+2*y+3*z+3*t=2,x-y+z=0,x+3*y-z+4*t=-2,3*x+4*y+3*
  z+7*t=0;
> A:=genmatrix([sistema],[x,y,z,t]);
> AM:=genmatrix([sistema],[x,y,z,t],flag);
> X:=matrix([[x],[y],[z],[t]]):
  evalm(A)*evalm(X)=evalm(b);
> [rank(A),rank(AM)];
```

Se trata de un sistema compatible indeterminado. Buscamos un sistema con submatriz de rango 3 e igual número de incognitas. Elegimos las tres primeras ecuaciones y pasamos la variable *t* al segundo miembro

```
> A1:= submatrix(A,1..3,1..3);
> det(%);
```

Esta submatriz es regular y por tanto contiene las incógnitas principales.

```
> b1:=matrix([[2-2*t],[0],[-2-4*t]]);
> X1:=matrix([[x],[y],[z]]); # incógnitas principales
> evalm(A1)*evalm(X1)=evalm(b1);
> AM1:=concat(A1,b1);
  [rank(A1),rank(AM1)];
> evalm(X1)=evalm(inverse(A1)&*b1);
```

Métodos directos de Maple

La orden **solve** de Maple resuelve el sistema directamente

solve({*ecu 1,ecu 2,...,ecu n***},{***incog 1,incog 2,...,incog m***});**

donde el primer argumento son las ecuaciones y el segundo las incógnitas.

Ejercicio: Resolver el sistema

$$\begin{gathered} x - y + z = 1 \\ 4x + 5y - 5z = 4 \\ 2x + y - z = 2 \\ x + 2y - 2z = 1 \end{gathered}$$

Resolución:

Apliquemos primero Rouché-Frobenius:

```
> sistema:=x-y+z=1,4*x+5*y-5*z=4,2*x+y-z=2,x+2*y-2*z=1;
> A:=genmatrix([sistema],[x,y,z]);
> AM:=genmatrix([sistema],[x,y,z],flag);
> [rank(A),rank(AM)];
```

Luego se trata de un sistema compatible con infinitas soluciones que me dependen de una incógnita. Resolvamos con el solve:

```
> solve({sistema},{x,y,z});
```

Diagonalización de matrices

El objetivo de esta sección es la diagonalización de matrices cuadradas. Es decir dada una matriz A, obtener dos matrices D (una matriz diagonal) y P (una matriz de paso) que permitan escribir la matriz A en función de las otras dos en la forma

$$A = PDP^{(-1)}$$

Dicha factorización nos permite calcular rápidamente potencias de A, esto es A^k , con **k** natural.

Para ello es necesario obtener los autovalores que son los que forman la matriz diagonal y los autovectores que son los que forman la matriz de paso, P.

Cálculo de autovalores y autovectores

Maple implementa comandos que permiten el trabajo fluido con autovalores y autovectores de una matriz cuadrada. Tenemos los siguientes comandos:

eigenvals(A)

Devuelve los autovalores de la matriz A(raíces del polinomio característico det(λId-A))

charpoly(A,variable)

Devuelve el polinomio característico de A en función de la variable dada, cuyo valor es det(variable* Id - A)

eigenvects(A)

Devuelve los autovectores de la matriz A. Además este comando nos da, aparte del autovector, el autovalor correspondiente junto a su multiplicidad.

Ejercicio.- Considerar la matriz cuadrada M de orden 2 cuyo elemento de lugar (i,j) viene definido por a(i,j) = $1/(\boldsymbol{i}^2+\boldsymbol{j}^2)$. Calcular sus autovalores y autovectores. Hallar el polinomio característico. Calcular sus raíces y comprobar que son los autovalores.

Solución

```
> restart:
> with(linalg):
> M:=matrix([[1/2,1/5],[1/5,1/8]]);
> eigenvals(M);
> p:=charpoly(M,lambda);
> solve(p=0,lambda);
> eigenvects(M);
```

Ejercicio.- Se dice que una matriz es definida positiva si todos los autovalores son positivos (positivo = mayor estricto que cero). Sean $A=\begin{bmatrix}1 & 3 & -5 & 7\\ 3 & 2 & 0 & 6\\ -5 & 0 & 1 & -1\\ 7 & 6 & -1 & 4\end{bmatrix}$ y $B=\begin{bmatrix}2 & 1 & 0 & 0\\ 1 & 2 & 1 & 0\\ 0 & 1 & 2 & 1\\ 0 & 0 & 1 & 2\end{bmatrix}$. ¿ Es A definida positiva? ¿ Es B definida positiva?

Solución

```
> restart:
> with(linalg):
> A := matrix([[1, 3, -5, 7], [3, 2, 0, 6], [-5, 0, 1, 
  -1], [7, 6, -1, 4]]);
> eigenvalues(A);
> evalf(allvalues(%));
> B := matrix([[2, 1, 0, 0], [1, 2, 1, 0], [0, 1, 2, 1], 
  [0, 0, 1, 2]]);
> eigenvalues(B);
> evalf(%);
```

¿Cuál es la conclusión?

Matriz diagonal y matriz de paso

El comando

jordan(A)

devuelve la forma cacónica de Jordan, es decir la matriz diagonal, que tiene a los aotovalores en la diagonal. Y
con el comando

jordan(A,'P')

devuelve la forma canónica de Jordan de la matriz A y la matriz de paso cuyas columnas son los autoavectores de A, cumpliéndose que

$P^{(-1)}\,A\,P = \mathbf{D}$

siendo P la matriz de paso, y D la matriz diagonal.
Para poder ver la matriz de paso es necesario añadir el comando

print(P)

Ejemplo: Dada la matriz $A = \begin{bmatrix} 3 & -2 & 4 \\ -2 & -2 & 6 \\ 4 & 6 & -1 \end{bmatrix}$. Calcular los autovalores, autovectores, matriz canónica (o diagonal) , matriz de paso y probar el resultado.

Solución

```
> A:=matrix([[3, -2, 4], [-2, -2, 6], [4, 6, -1]]);
  eigenvals(A);
  eigenvects(A);
  D1:=jordan(A);
  jordan(A,'P');
  print(P);
> evalm(P&*D1&*P^(-1))=evalm(A);
```

Ejemplo: Dada la matriz $A = \begin{bmatrix} 1 & -1 & 3 \\ -1 & 1 & 2 \\ 1 & 1 & 2 \end{bmatrix}$. Calcular los autovalores, autovectores, matriz canónica (o diagonal) , matriz de paso y probar el resultado.

Solución

```
> restart:
> with(linalg):
> A:=matrix([[1,-1,3],[-1,1,2],[1,1,2]]);
  eigenvals(A);
  eigenvects(A);
  D2:=jordan(A);
  jordan(A,'P');
  print(P);
> evalf(evalm(P&*D2&*P^(-1)))=evalm(A);
```