



PRÁCTICA Nº 13

APLICACIONES LINEALES II

1. OPERACIONES CON APLICACIONES LINEALES Y RELACIÓN CON LAS MATRICES ASOCIADAS.

Dados V y V' dos espacios vectoriales sobre un cuerpo K , denotaremos por $\text{Hom}_K(V, V')$ al conjunto de todas las aplicaciones lineales de V en V' . En este conjunto se podemos definir operaciones suma y producto por escalar de la forma:

Dadas $f, g \in \text{Hom}_K(V, V')$ y $\lambda \in K$ se define las aplicaciones lineales:

$$\begin{aligned} f + g: V &\rightarrow V'; (f + g)(u) = f(u) + g(u) \\ \lambda f: V &\rightarrow V'; (\lambda f)(u) = \lambda f(u) \end{aligned}$$

Dadas aplicaciones lineales $f: V \rightarrow V'$ y $g: V' \rightarrow V''$, su composición $g \circ f: V \rightarrow V''$ definida por $(g \circ f)(x) = g(f(x))$ es también lineal.

Veamos cómo la asignación a una aplicación lineal de su matriz asociada se comporta bien respecto a las operaciones con aplicaciones lineales:

Proposición. Sean V, V' y V'' espacios vectoriales sobre K de dimensiones finitas, B, B' y B'' bases de V, V' y V'' respectivamente y $f, g: V \rightarrow V'$ y $h: V' \rightarrow V''$ aplicaciones lineales, entonces se tiene:

1. $M_{B,B'}(f + g) = M_{B,B'}(f) + M_{B,B'}(g)$.
2. $M_{B,B'}(\lambda f) = \lambda M_{B,B'}(f)$, para todo $\lambda \in K$.
3. $M_{B,B''}(h \circ f) = M_{B,B''}(h) M_{B,B'}(f)$.

■

Ejemplo. Calcular las matrices asociadas a f, g y h respecto de las bases canónicas y comprobar la proposición anterior, siendo:

$$f: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \text{ dada por } f(x, y, z) = (x + y, 3x + y - z, y + 5z).$$

$$g: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \text{ dada por } g(x, y, z) = (2x, y + z, x + y).$$

$h: \mathbb{R}^3 \rightarrow \mathbb{R}^4$ dada por $h(x, y, z) = (2x, x + y, 3x + y - z, 2y + z)$.

```
In[ ]:=      f[{x_,y_,z_}] := {x+y, 3x+y-z, y+5z}
             g[{x_,y_,z_}] := {2x, y+z, x+y}
             h[{x_,y_,z_}] := {2x-z, x+y, 3x+y-z, 2y+z}
             s[{x_,y_,z_}] = f[{x,y,z}] + g[{x,y,z}];
             p[{x_,y_,z_}] = 3*f[{x,y,z}];
             c[{x_,y_,z_}] = h[f[{x,y,z}]];
             B= IdentityMatrix[3];
             Af = Transpose[Table [f[B[[i]]],{i,1,3}]];
             Ag = Transpose[Table [g[B[[i]]],{i,1,3}]];
             Ah = Transpose[Table [h[B[[i]]],{i,1,3}]];
             As = Transpose[Table [s[B[[i]]],{i,1,3}]];
             Ap = Transpose[Table [p[B[[i]]],{i,1,3}]];
             Ac = Transpose[Table [c[B[[i]]],{i,1,3}]];
```

```
In[ ]:=      Af + Ag ==As
```

```
Out[ ]:=     True
```

```
In[ ]:=      3*Af ==Ap
```

```
Out[ ]:=     True
```

```
In[ ]:=      Ah.Af ==Ac
```

```
Out[ ]:=     True
```



2. NÚCLEO E IMAGEN DE UNA APLICACIÓN LINEAL

Dados V y V' dos espacios vectoriales sobre K y $f: V \rightarrow V'$ una aplicación lineal, se definen el **núcleo de f** como el subespacio de V dado por:

$$\text{Ker}(f) = \{x \in V \mid f(x) = 0\}$$

y se define la **imagen de f** como el subespacio de V' dado por:

$$\text{Im}(f) = \{f(x) \mid x \in V\}$$

Un primer método para calcular el núcleo, conocida la expresión matricial de f , es decir, $Y = AX$, consistiría en resolver el sistema homogéneo que resulta de plantear $f(x) = 0$, es decir, $AX = 0$. Si el sistema anterior es S.C.D. entonces $\text{Ker}(f) = \{0\}$ y si es un S.C.I. entonces $\text{Ker}(f) = L(\{u_1, u_2, \dots, u_r\})$ siendo $\{u_1, u_2, \dots, u_r\}$ una base del subespacio vectorial de soluciones del sistema $AX = 0$. La orden `NullSpace` nos permite obtener esta base directamente.

Ejemplo. Calculamos la base, dimensión, ecuaciones paramétricas e implícitas del núcleo de la aplicación lineal $f: \mathbb{R}^3 \longrightarrow \mathbb{R}^4$ dada por $f(x, y, z) = (2x-z, x+y, 3x+y-z, 2y+z)$.

En primer lugar, calculemos la matriz asociada a la aplicación lineal respecto de las bases canónicas:

```
In[ ]:=      f[{x_,y_,z_}] := {2x-z, x+y, 3x+y-z, 2y+z}
             B= IdentityMatrix[3];
             A= Transpose[Table[f[B[[i]]],{i,1,3}]];
             MatrixForm[A]
```

```
Out[ ]=      
$$\begin{pmatrix} 2 & 0 & -1 \\ 1 & 1 & 0 \\ 3 & 1 & -1 \\ 0 & 2 & 1 \end{pmatrix}$$

```

Para calcular el núcleo podemos resolver el sistema de ecuaciones $AX = 0$ con `Solve[]`, que nos proporciona sus ecuaciones implícitas:

```
In[ ]:=      Solve[A.{x1,x2,x3}=={0,0,0,0},{x1,x2,x3}]
```

```
Out[ ]=      {{x2 -> -x1, x3 -> 2x1}}
```

O podemos calcular la base del núcleo:

```
In[ ]:=      baseNucleo=NullSpace[A]
```

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

Usando lo aprendido en la práctica de subespacios vectoriales, podríamos calcular la base y las ecuaciones paramétricas desde las ecuaciones implícitas; o las ecuaciones paramétricas e implícitas desde la base.

También podemos hacerlo usando la función `LogicalExpand[]` de Mathematica. Para ello, observemos que la dimensión del núcleo es 1 y necesitaremos un **parámetro**, a , para el cálculo de las paramétricas. Como el núcleo es un subespacio de \mathbb{R}^3 , escribiremos una lista con tres **coordenadas** $\{x,y,z\}$ y formaremos una ecuación de estas coordenadas, con la lista formada por las coordenadas de los vectores de la base del núcleo, multiplicada matricialmente por la lista de los parámetros. Por último, la función **LogicalExpand** igualará término a término las listas implicadas, dando lugar a las ecuaciones paramétricas en la forma habitual.

```
In[ ]:=      param={a};
             coord={x,y,z};
             paramNucleo=LogicalExpand[coord ==
             Transpose[baseNucleo].param]
```

```
Out[ ]=      x == a && y == -a && z == 2a
```

La orden **Eliminate[paramNucleo, param]** hace que se elimine el único parámetro que hay en este caso, obteniendo las ecuaciones implícitas:

```
In[ ]:= Eliminate[paramNucleo, param]
```

```
Out[ ]= x == -y && 2y == -z
```

Recordemos que el **número de ecuaciones implícitas** de un subespacio U de un espacio vectorial V , es igual a $\dim(V) - \dim(U)$. En nuestro ejemplo, efectivamente, nos han salido $3 - 1 = 2$ ecuaciones implícitas del núcleo de f .

Para calcular la imagen de la aplicación lineal, buscaremos un sistema de generadores, que según vimos en una proposición, puede obtenerse a partir de los transformados mediante f de cualquier sistema de generadores del dominio. Teniendo en cuenta esto, sabemos que las columnas de la matriz asociada a f , permiten obtener un sistema generador de $\text{Im}(f)$. Así, una base no será más que el conjunto formado por el mayor número de columnas que sean linealmente independientes y que podemos obtenerlas a partir de la forma normal de Hermite de la matriz asociada a f .

Ejemplo. Calculamos la base, dimensión, ecuaciones paramétricas e implícitas de la imagen de la aplicación lineal $f: \mathbb{R}^3 \longrightarrow \mathbb{R}^4$ dada por $f(x, y, z) = (2x-z, x+y, 3x+y-z, 2y+z)$.

Como se trata de la misma aplicación lineal del ejemplo anterior, ya tenemos la matriz asociada a la aplicación lineal respecto de las bases canónicas, A .

```
In[ ]:= generadorImagen=Transpose[A];  
RowReduce[generadorImagen]
```

```
Out[ ]= {{1, 0, 1, -1}, {0, 1, 1, 2}, {0, 0, 0, 0}}
```

Como vemos, en este caso los linealmente independientes son las dos primeras filas, por tanto, la base será:

```
In[ ]:= baseImagen=Table[%[[i]],{i,2}]
```

```
Out[ ]= {{1, 0, 1, -1}, {0, 1, 1, 2}}
```

Igual que con el núcleo, podemos usar lo aprendido en la práctica de subespacios vectoriales para determinar las ecuaciones paramétricas e implícitas. O también, podemos hacerlo, introduciendo dos parámetros $\{a,b\}$, porque es de dimensión 2, y 4 coordenadas $\{x,y,z,t\}$, porque la imagen es subespacio vectorial de $V' = \mathbb{R}^4$. Las ecuaciones correspondientes son:

```
In[ ]:= param={a,b};  
coord={x,y,z,t};  
paramImagen=LogicalExpand[coord ==  
Transpose[baseImagen].param]
```

`Out[]:=` $t = -a+2b$ && $x = a$ && $y = b$ && $z = a+b$

`In[]:=` **Eliminate[paramNucleo, param]**

`Out[]:=` $t = -x+2y$ && $x = -y+z$

En este caso hemos obtenido $\dim V^* - \dim \text{Im}(f) = 4 - 2 = 2$, por tanto, 2 ecuaciones implícitas del subespacio imagen de la aplicación lineal f .

Como fácilmente se puede observar, el método anterior no es totalmente programable, pues es necesario intervenir añadiendo los parámetros y coordenadas necesarios para las ecuaciones paramétricas, o eliminando las filas nulas para la base de la imagen. Veamos como la forma normal de Hermite nos facilita el cálculo del núcleo y la imagen de f .

Si al calcular la forma de Hermite por columnas de A realizamos las operaciones elementales sobre $\begin{pmatrix} A \\ I \end{pmatrix}$, obtenemos $\begin{pmatrix} C \\ P \end{pmatrix}$ donde P es la matriz regular de orden n tal que $C = A.P$, pues bien se tiene que las columnas no nulas de C forma una base de $\text{Im}(f)$ y las columnas de P que están bajo las columnas de ceros de C (si las hay) forman una base de $\text{Ker}(f)$.

Recordemos que en el Mathematica la orden **RowReduce[A]** nos calcula la forma normal de Hermite por filas, luego al hacer lo anterior con el Mathematica nosotros trabajaremos por filas transponiendo la matriz $\begin{pmatrix} A \\ I \end{pmatrix}$ antes de calcular la forma de Hermite y al final transponiendo el resultado $(C | P)$.

Ejemplo. Calcular el núcleo y la imagen de la aplicación lineal $f: \mathbb{R}^3 \longrightarrow \mathbb{R}^4$ dada por $f(x, y, z) = (2x-z, x+y, 3x+y-z, 2y+z)$ usando para ello la forma normal de Hermite.

SOLUCIÓN:

```
In[]:= f[{x_,y_,z_}] := {2x-z,x+y,3x+y-z,2y+z}
B= IdentityMatrix[3];
Bp= IdentityMatrix[4];
A= Transpose[Table[LinearSolve[Transpose[Bp], f[B[[i]]]],{i,1,3}]];
Join[A,B];
AI=Transpose[%];
CP=RowReduce[AI];
MatrixForm[Transpose[CP]]
```

`Out[]:=`

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

Por tanto, una base de la imagen es $\{(1,0,1,1),(0,1,1,-2)\}$ y una base del núcleo es $\{(1,-1,2)\}$.

2.2 TIPOS DE APLICACIONES LINEALES.

Una aplicación lineal pueden ser **monomorfismo, epimorfismo o isomorfismo** si como aplicación es inyectiva, sobreyectiva o biyectiva, respectivamente.

Proposición

Dada una aplicación lineal $f: V \longrightarrow V'$, $\dim(V)=n$, $\dim(V')=m$, se verifica:

1. f es inyectiva $\Leftrightarrow \text{Ker}(f) = 0 \Leftrightarrow \text{rang}(A) = n$.
2. f es sobreyectiva $\Leftrightarrow \text{Im}(f) = V' \Leftrightarrow \text{rang}(A) = m$.
3. f es biyectiva $\Leftrightarrow A$ es cuadrada y regular.

Según la proposición anterior la aplicación f del ejemplo no es ni inyectiva pues, $\dim(\text{Ker}(f))=1$, ni sobreyectiva, pues $\dim(\text{Im}(f))=2$. Por tanto, f tampoco es biyectiva.