NOMBRE:

PRÁCTICA 3. APLICACIONES LINEALES, PRODUCTO ESCALAR Y RESOLUCIÓN DE ECUACIONES.

1 *INTRODUCCIÓN.*

El alumno debe ir realizando todos los EJEMPLOS que aparecen en esta práctica e ir anotando los resultados en la hoja que se le ha entregado. Además de estos ejemplos, hay planteados unos EJERCICIOS, que, a diferencia de lo que ocurre con los EJEMPLOS, no vienen desarrollados en el lenguaje Maxima, sino que tendrá que ser el alumno el que tenga que introducirlos para obtener el resultado.

2 *APLICACIONES LINEALES.* *SOLUCIÓN.*

2.1 Matriz asociada a una aplicación lineal.

```
--> /*EJEMPLO 1: Introducir la matriz asociada a la a. lineal
    f:R^4->R^3 dada por f(x,y,z,t)=(x-z,0,y-2z+t)*/

    f(x,y,z,t):=([x-z,0,y-2*z+t]);
    B:apply(matrix,[f(1,0,0,0),f(0,1,0,0),f(0,0,1,0),f(0,0,0,1)]);
    A:transpose(B);
```

```
--> /*EJEMPLO 1 BIS: Introducir la matriz asociada a la a. lineal
    f:R^4->R^3 dada por f(x,y,z,t)=(x-z,0,y-2z+t)*/

    A:matrix([1,0,-1,0],[0,0,0,0],[0,1,-2,1]);
```

2.2 Imagen y Núcleo de una aplicación lineal.

```
--> /*EJEMPLO 2: Vamos a hallar el subespacio Im(f), siendo
    f la apl. lineal anterior*/

    columnspace(A);
    args(%);
```



```
--> /*EJEMPLO 3: Vamos a hallar el subespacio ker(f), siendo
    f la apl. lineal anterior*/

    nullspace(A);
    args(%);
```

2.3 Cambio de base en una aplicación lineal.

```
--> /*EJEMPLO 4: Vamos a hallar la nueva matriz asociada a f
    cuando se considera en R^4 la base canónica y en R^3 la
    nueva base dada por B={(1,1,0),(1,1,1),(1,0,0)}*/
    A:matrix([1,0,-1,0],[0,0,0,0],[0,1,-2,1]);
    MBC_id:matrix([1,1,1],[1,1,0],[0,1,0]);
    MBC_id.A;
```

EJERCICIO 1: Dada la aplicación lineal f:R^3-->R^2 definida por
f(x,y,z)=(x-y-z,x-y-z)
a) Escribir la matriz asociada a f en las bases canónicas.
b) Hallar bases para Im(f) y ker(f). Clasificar f.
c) Hallar la nueva matriz asociada a f cuando se consideran las nuevas bases B1 y B2 dadas por
B1={(1,1,0),(1,1,1),(1,0,0)} y B2={(2,-1),(3,1)}

3 *PRODUCTO ESCALAR. ESPACIO VECTORIAL EUCLÍDEO.*

```
--> /*EJEMPLO 5: Calculamos el producto escalar de los vectores
    (1,2,3) y (-1,5,0)*/

    [1,2,3].[-1,5,0];
```

```
--> /*EJEMPLO 6: Calculamos el módulo de los vectores
    v=(1,2,3) y w=(-1,5,0)*/

    v:[1,2,3]$
    w:[-1,5,0]$
    modulo_v:(v.v)^(1/2);
    modulo_w:(w.w)^(1/2);
```

3.1 El método de Gram-Schmidt.

```
--> /*EJEMPLO 7: Usar el método de G-S para ortogonalizar los
    vectores {(1,2,3),(9,18,30),(12,48,60)}*/

    load(eigen)$
    x:matrix([1,2,3],[9,18,30],[12,48,60]);
    y:gramschmidt(x);
```

```
--> /*EJEMPLO 8*/

    load(eigen)$
    ip(f,g):=integrate(f*g,u,a,b);
    y:gramschmidt([1,sin(u),cos(u)],ip),a=-%pi/2,b=%pi;
```

EJERCICIO 2: Comprobar que, con el producto escalar definido por ip(f,g) del ejemplo anterior, las funciones que se obtienen en el mismo efectivamente son ortogonales a las funciones {1,sin(u),cos(u)}.

EJERCICIO 3: Calcular (usando la definición de producto escalar anterior) el módulo de cada uno de los vectores obtenidos en el ejemplo 8, y construir la correspondiente base ortonormal.

4 RESOLUCIÓN DE ECUACIONES Y SISTEMAS (no lineales).

4.1 Soluciones de ecuaciones algebraicas.


```
--> /*EJEMPLO 9: Resolvemos la ecuación x^4-5x^2-3=0*/

    kill(all)$ /*Con este comando eliminamos todos los valores
    que hayamos asignado a variables o funciones anteriormente*/
    solve(x^4-5*x^2-3=0,x);
```

```
--> /*EJEMPLO 10: Podemos resolver algunas ecuaciones de grado
    mayor que 4, como por ejemplo x^6-1=0*/

    solve(x^6-1=0,x);
```

```
--> /*EJEMPLO 11: Intentamos resolver x^5-4x+2=0*/

    solve(x^5-4*x+2=0,x);
```

```
--> /*EJEMPLO 12: Resolver numéricamente x^5-4x+2=0*/

    allroots(x^5-4*x+2=0);
```

```
--> /*EJEMPLO 12 BIS: Calcular todas las raíces de x^5-6x^2+8x+3=0
    y determinar las que son reales*/

    allroots(x^5-6*x^2+8*x+3=0);
    realroots(x^5-6*x^2+8*x+3=0);
```

```
--> /*EJEMPLO 13: Resolver el sistema lineal
    ax+y=0; 2x+(1-a)y=1*/

    solve([a*x+y=0,2*x+(1-a)*y=1],[x,y]);
```

```
--> /*EJEMPLO 14: Resolver el sistema no lineal
    x^2+xy+y^2=4; x+xy+y=2*/

    solve([x^2+x*y+y^2=4,x+x*y+y=2],[x,y]);
```

```
--> /*EJEMPLO 15: Podemos obtener una mejor presentación de la
    solución anterior*/

    %,rectform;
```

EJERCICIO 4:
a) Hallar las raíces de x^3-3x+1=0 mediante solve.

b) Hallar todas las raíces del polinomio x^5-2x^3+x^2-1.

c) Resolver el sistema lineal -3x+2y=3; x-4y=-1.

d) Resolver el sistema no lineal 3x^2+2y=0; x+2y^2-y=-1.

4.2 Soluciones de ecuaciones trascendentes.

```
--> load(to_poly_solver)$
```

```
--> /*EJEMPLO 16: Podemos resolver la ecuación algebraica
    2x^2-3x+5=0 usando este comando*/

    to_poly_solve(2*x^2-3*x+5=0,x);
```

```
--> /*EJEMPLO 17: Podemos resolver el sistema de ec. lineales
    2x-y=5; x+3y=1 usando este comando*/

    to_poly_solve([2*x-y=5,x+3*y=1],[x,y]);
```

```
--> /*EJEMPLO 18: Podemos resolver un sistema no lineal, como
    (x^2+1)(y^2+1)=10; (x+y)(xy-1)=3*/

    to_poly_solve([(x^2+1)*(y^2+1)=10,(x+y)*(x*y-1)=3],[x,y]);
```

```
--> /*EJEMPLO 19: Intentamos resolver el sistema
    x^2+y^2=4; (x-1)^2+(y-1)=4*/

    to_poly_solve([x^2+y^2=4,(x-1)^2+(y-1)^2=4],[x,y]);
```



```
--> /*EJEMPLO 19 BIS: Intentamos resolver el sistema
    x^2+y^2=4; (x-1)^2+(y-1)=4*/

    to_poly_solve([x^2+y^2=4,(x-1)^2+(y-1)^2=4],[x,y],
    use_grobner=true);
```

```
--> /*EJEMPLO 20: Intentamos resolver exp(x^2-1)-x = 0*/

    to_poly_solve(exp(x^2-1)-x = 0,x);
```

EJERCICIO 5: Resolver las siguientes ecuaciones:
a) (x+x^(1/2))^(1/2)-(x-x^(1/2))^(1/2) = 3/2 (x/(x+x^(1/2)))^(1/2)

b) log(1+(x+1)^(1/2)) = log(x-40)

c) (cos(x))/(sin(x)+1) + (sin(x)+1)/(cos(x)) = 4

d) sin(x + Pi/6) = cos(x - Pi/4)

e) tan(x) = cotan(x)

f) sin(x^2+1) = cos(x)