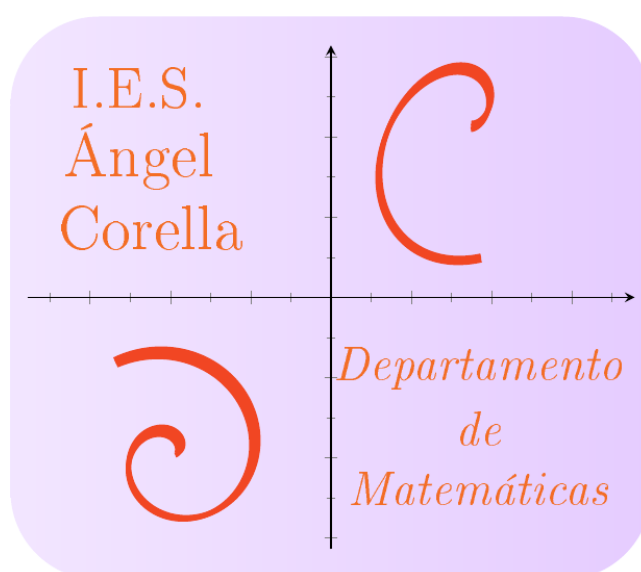


WxMaxima para Bachillerato

David Matellano

27 de abril de 2022



Índice

1. Introducción de constantes y funciones	1
1.1. Constantes especiales	2
2. Funciones	3
2.1. Realización de logaritmos con Maxima	4
2.1.1. Logaritmos en cualquier base	4
2.2. Algunas funciones de usuario útiles en Maxima:	5
3. Aplicamos lo aprendido	5
4. Matrices y determinantes	7
4.1. Introducción de una matriz	7
4.2. Operaciones básicas con matrices	8
4.2.1. Escalar por una matriz	8
4.2.2. Suma y resta de matrices	9
4.2.3. Producto de matrices	9
4.2.4. Potencia de una matriz	10
4.3. Traspuesta de una matriz	11
4.4. Matriz adjunta	11
4.5. Matriz inversa	12
5. Determinante de una matriz	13
6. Submatrices de una matriz	13
7. Sistemas lineales con Maxima	15
7.1. Resolver el sistema lineal	15
7.2. Matrices del sistema	17
7.3. Discusión de un sistema	18
7.3.1. Rango de una matriz	18
7.4. Sistemas dependientes de algún parámetro	18
7.5. Triangularizar matrices: El método de Gauss	23
8. Inecuaciones	24
8.1. Librerías y funciones <i>mías</i>	24
8.2. Inecuaciones de una variable	24
8.2.1. Inecuaciones de primer grado	24
8.2.2. Inecuaciones de 2º grado	25

8.2.3. Inecuaciones polinómicas	25
8.2.4. Inecuaciones con valores absolutos	26
8.2.5. Inecuaciones racionales	27
8.3. Sistemas de inecuaciones de una variable	27
8.4. Inecuaciones de dos variables	28
8.5. Recintos	29
9. Vectores	35
9.1. Cargar una librería necesaria	35
9.2. Funciones propias útiles para vectores	35
9.3. Introducción de vectores	36
9.4. Operaciones básicas con vectores	36
9.5. Producto escalar	36
9.6. Módulo de un vector	36
9.7. Vectores unitarios	37
9.8. Producto vectorial	37
9.9. Producto mixto	38
10. Geometría espacial con Maxima	39
10.1. Rectas en el espacio	39
10.2. Planos	41
10.3. Intersecciones	42
10.3.1. Intersección entre dos rectas	42
10.3.2. Intersección recta-plano	44
10.3.3. Intersección entre dos planos	45
10.4. Distancias	45
10.4.1. Distancia entre dos puntos	46
10.4.2. Distancia de un punto a una recta	46
10.4.3. Distancia de un punto a un plano	47
11. Configurar arranque de Maxima	49

Resumen

Pequeño manual de las utilidades más importantes para los alumnos de Bachillerato, en las asignaturas de Matemáticas y Física.

1. Introducción de constantes y funciones

☞ Podemos asignar el valor de una constante mediante el uso de los dos puntos:



```
(% i2) a:3;
      b:5;
```

```
(a)                                     3
```

```
(b)                                     5
```

Ahora podemos operar con ellas:



```
(% i3) d=a+2*b^2;
```

```
(% o3)                                     d = 53
```

Nótese que hemos puesto un igual en lugar de los dos puntos, por lo que d no tiene asignado ningún valor:



```
(% i4) d;
```

```
(% o4)                                     d
```

Si queremos que quede registrado, asignamos de nuevo con :



```
(% i5) d:2*a+3*b;
```

```
(d)                                     21
```

```
(% i6) 3*d;
```

```
(% o6)                                     63
```

A veces necesitamos liberar una constante. Para ello, utilizamos el siguiente comando:



(% i7) remvalue(a,b);

(% o7) $[a, b]$

(% i10) a;b;d;

(% o8) a

(% o9) b

(% o10) 21

1.1. Constantes especiales

Algunas constantes especiales se introducen con el símbolo % delante. Ejemplos



(% i13) %pi;%e;%i /* unidad imaginaria */;

(% o11) π

(% o12) %e

(% o13) %i

Ejemplos de su uso:



(% i14) A=3*sin(%pi/6);

(% o14) $A = \frac{3}{2}$

(% i23) B=%e^0-1;

(% o23) $B = 0$

2. Funciones

Creamos funciones de una o más variables con los caracteres := Veamos un ejemplo:



(% i15) f(x):=3*x^2+1;

(% o15)

$$f(x) := 3x^2 + 1$$

Podemos así, por ejemplo, obtener el valor numérico $f(8)$ y asignárselo a una constante a:



(% i16) a:f(8);

(a)

193

Otro ejemplo:



(% i17) g(x,y):=(x+y)^2-x^3+y^3;

(% o17)

$$g(x, y) := (x + y)^2 - x^3 + y^3$$

(% i18) g(2,-1);

(% o18)

-8

2.1. Realización de logaritmos con Maxima

☞ El uso de la función logarítmica ha de realizarse con cuidado en Maxima, ya que sólo tiene implementado el logaritmo neperiano y con la sintaxis $\log(x)$.

Si introducimos $\ln(\%e)$, no devuelve 1, ya que no está realizando dicho logaritmo. Veamos:



```
(% i20) ln(%e);
```

```
(% o20) ln(%e)
```

☞ La manera correcta de realizar el $\ln(e)$ será:



```
(% i21) log(%e);
```

```
(% o21) 1
```

2.1.1. Logaritmos en cualquier base

Para realizar el logaritmo en cualquier base, basta recordar la propiedad de los cambios de bases de los logaritmos:

$$\log_a(x) = \frac{\ln(x)}{\ln(a)}$$

Así, podemos implementar una función $\log_a(x,a)$ para realizar $\log_a x$:



```
(% i22) loga(x,a):=log(x)/log(a);
```

```
(% o22) loga(x,a) := log(x) / log(a)
```

Veamos su uso calculando $\log(10000)$



```
(% i23) loga(10000,10),numer;
```

```
(% o23) 4.0
```

2.2. Algunas funciones de usuario útiles en Maxima:

Yo utilizo algunas funciones muy útiles para matemáticas: Transformar grados en radianes y viceversa:



(% i24) `gra(x):=180*x/%pi /* Nótese cómo introducimos el número pi */;`

(% o24)
$$\text{gra}(x) := \frac{180x}{\pi}$$

(% i25) `gra(%pi/3);`

(% o25)
$$60$$

(% i26) `rad(x):=%pi*x/180;`

(% o26)
$$\text{rad}(x) := \frac{\pi x}{180}$$

(% i27) `rad(60);`

(% o27)
$$\frac{\pi}{3}$$

3. Aplicamos lo aprendido

Vamos a ver cómo podemos aplicar lo aprendido hasta aquí para resolver dos pequeños ejercicios de Física:

Ejercicio 1

Calcula la aceleración de la gravedad en la superficie terrestre.:

Datos: masa de la Tierra, $M_t = 5,97 \cdot 10^{24} \text{ kg}$; constante de gravitación universal, $G = 6,67 \cdot 10^{-11} \text{ Nm}^2/\text{kg}^2$; radio terrestre, $R_t = 6370 \text{ km}$



Recuerda la expresión de $|\vec{g}|$:

$$|\vec{g}| = \frac{G \cdot M}{r^2}$$

☞ Introducimos los datos y operamos



```
(% i30) Mt:5.97e24;
      G:6.67e-11;
      Rt:6370e3 /* m */;
```

```
(Mt) 5.9710^24
```

```
(G) 6.6710^-11
```

```
(Rt) 6370000.0
```

```
(% i31) g=G*Mt/Rt^2;
```

```
(% o31) g = 9.813440652193736
```

Ejercicio 2

Aplica la Ley de Snell para hallar \hat{r} si $n_1 = 1,5$, $n_2 = 1,33$ e $\hat{i} = 30^\circ$



La Ley de Snell respecto a la refracción de un rayo de luz es:

$$n_1 \operatorname{sen} \hat{i} = n_2 \operatorname{sen} \hat{r}$$

Si despejamos \hat{r} obtenemos: $\hat{r} = \operatorname{arcsen} \left(\frac{n_1 \operatorname{sen} \hat{i}}{n_2} \right)$

Vamos a calcularlo con WxMaxima, haciendo uso de las funciones para pasar de grados a radianes y viceversa vistas en el apartado 2.2, ya que Maxima utiliza radianes para las funciones trigonométricas. Añadimos el comando ,numer para que de la expresión decimal del resultado:



```
(% i34) n1:1.5;
      n2:1.33;
      i:30 /* grados */;
```

```
(n1) 1.5
```

```
(n2) 1.33
```

```
(i) 30
```

```
(% i35) r:gra(asin(n2*sin(rad(i))/n1)),numer;
```

```
(r) 26.31675512821711
```

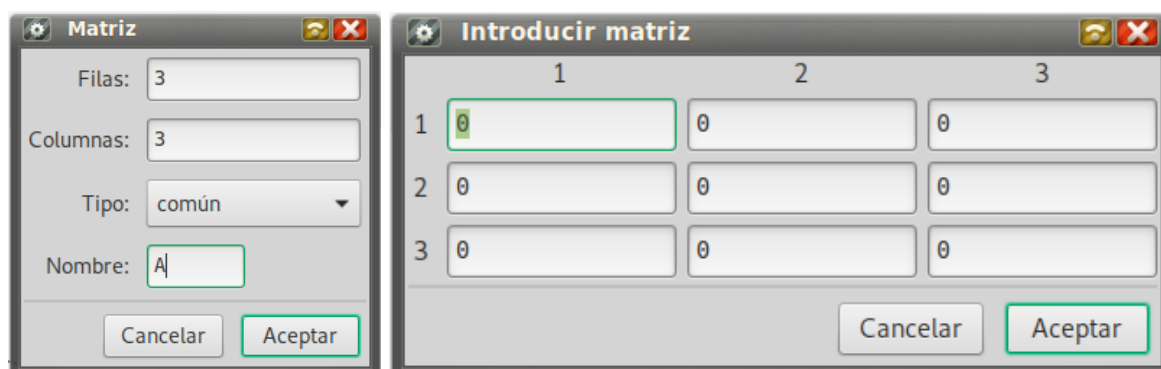
4. Matrices y determinantes

Vamos a ver que esta es una de las herramientas más potentes de Maxima para el estudiante de 2º de Bachillerato, en cualquiera de sus modalidades.

4.1. Introducción de una matriz

Podemos introducir cualquier matriz desde la línea de comandos, pero WxMaxima cuenta con una sencilla herramienta para introducir matrices.

Para ello, la invocaremos a través del menú **Álgebra** → **Introducir matriz**, definimos sus dimensiones, tipo y nombre (figura 1(a)) e introducimos sus valores (figura 1(b))



(a) Menú matriz

(b) Valores de la matriz

Figura 1: Figuras de cómo introducir una matriz

Introduzcamos, por ejemplo las siguientes matrices:

$$A = \begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}; \quad B = \begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 2 & -3 \end{pmatrix}$$

Tras introducirlas con el menú anterior, se puede ver además cómo introducirlas a través de comandos, lo cual puede ser útil a la hora de modificar alguno de los valores de la matriz:



(% i36) A: matrix([2,1,3],[-1,3,4],[1,1,2]);

(A)
$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% i37) B: matrix([1,2],[-1,0],[2,-3]);

(B)
$$\begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 2 & -3 \end{pmatrix}$$

4.2. Operaciones básicas con matrices

Veamos cómo realizar las operaciones con matrices: Para ello, vamos a introducir además otra matriz $C_{3 \times 3}$ para poder realizar sumas y restas de matrices:



(% i38) C: matrix([3,4,-1],[2,4,-2],[7,-1,1]);

(C)
$$\begin{pmatrix} 3 & 4 & -1 \\ 2 & 4 & -2 \\ 7 & -1 & 1 \end{pmatrix}$$

Comencemos:

4.2.1. Escalar por una matriz

Vamos a obtener una matriz $D = 2 \cdot A$. Para ello utilizamos el asterisco * para multiplicar:



(% i39) D:2*A;

(D)

$$\begin{pmatrix} 4 & 2 & 6 \\ -2 & 6 & 8 \\ 2 & 2 & 4 \end{pmatrix}$$

4.2.2. Suma y resta de matrices

Realicemos $D = 3A - 4C$



(% i40) D:3*A-4*C;

(D)

$$\begin{pmatrix} -6 & -13 & 13 \\ -11 & -7 & 20 \\ -25 & 7 & 2 \end{pmatrix}$$

4.2.3. Producto de matrices

Este es uno de los puntos singulares de Maxima: Observemos la salida del producto $A * C$



(% i42) A;C;

(% o41)

$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% o42)

$$\begin{pmatrix} 3 & 4 & -1 \\ 2 & 4 & -2 \\ 7 & -1 & 1 \end{pmatrix}$$

(% i43) A*C;

(% o43)

$$\begin{pmatrix} 6 & 4 & -3 \\ -2 & 12 & -8 \\ 7 & -1 & 2 \end{pmatrix}$$

☞ El asterisco no realiza el producto de matrices, sino que realiza el producto $a_{ij} \cdot c_{i,j}$, que NO ES COMO SE REALIZA UN PRODUCTO DE MATRICES

☞ Para realizar el producto correctamente, utilizamos el punto normal de escritura:



(% i44) A.C;

(% o44)

$$\begin{pmatrix} 29 & 9 & -1 \\ 31 & 4 & -1 \\ 19 & 6 & -1 \end{pmatrix}$$

Así, para realizar $D = 2A \cdot 3C$, haremos:



(% i45) D:(2*A).(3*C);

(D)

$$\begin{pmatrix} 174 & 54 & -6 \\ 186 & 24 & -6 \\ 114 & 36 & -6 \end{pmatrix}$$

4.2.4. Potencia de una matriz

Para elevar una matriz a una potencia, hay que utilizar el símbolo \wedge **dos veces**, ya que una única vez realiza el cuadrado de cada término a_{ij} de la matriz. Veamos un ejemplo, en el que tendremos la matriz A , la matriz $A1 = a_{ij}^2$, y la matriz correcta A^2 :



(% i48) A;A1=A^2;A^^2;

(% o46)

$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% o47)

$$A1 = \begin{pmatrix} 4 & 1 & 9 \\ 1 & 9 & 16 \\ 1 & 1 & 4 \end{pmatrix}$$

(% o48)

$$\begin{pmatrix} 6 & 8 & 16 \\ -1 & 12 & 17 \\ 3 & 6 & 11 \end{pmatrix}$$

4.3. Traspuesta de una matriz

Lo podemos hacer gráficamente a través del menú **Álgebra** → **Transponer matriz**. Si no hemos seleccionado previamente ninguna matriz, lo hará sobre la última salida de Maxima (%). Ello introducirá el comando **transpose**, que es la manera de trasponeer matrices en Maxima. Veamos un par de ejemplos:



(% i49) D:(2*A).(3*C);

(D)
$$\begin{pmatrix} 174 & 54 & -6 \\ 186 & 24 & -6 \\ 114 & 36 & -6 \end{pmatrix}$$

(% i50) transpose(%);

(% o50)
$$\begin{pmatrix} 174 & 186 & 114 \\ 54 & 24 & 36 \\ -6 & -6 & -6 \end{pmatrix}$$

(% i52) B;transpose(B);

(% o51)
$$\begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 2 & -3 \end{pmatrix}$$

(% o52)
$$\begin{pmatrix} 1 & -1 & 2 \\ 2 & 0 & -3 \end{pmatrix}$$

4.4. Matriz adjunta

De nuevo, en este apartado hay que tener cuidado, ya que **Maxima utiliza la notación anglosajona**, en la que llaman matriz adjunta a *nuestra* matriz adjunta traspuesta. Para ello, lo más útil es crear un función que vuelva a trasponeer a la salida del comando **adjoint**, que dará la matriz adjunta traspuesta. Veámoslo con un ejemplo:



(% i54) A;Ad=adjoint(A);

(% o53)
$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% o54)
$$Ad = \begin{pmatrix} 2 & 1 & -5 \\ 6 & 1 & -11 \\ -4 & -1 & 7 \end{pmatrix}$$

☞ Si el lector se toma la molestia de comprobarlo, verá que la salida obtenida es la matriz **adjunta traspuesta**. Para hacerlo correctamente, definamos una función **adjunta**:



```
(% i55) adjunta(A):=transpose(adjoint(A));
```

```
(% o55) adjunta(A) := transpose (adjoint(A))
```

```
(% i57) Amal=adjoint(A);Ad=adjunta(A);
```

```
(% o56) Amal =  $\begin{pmatrix} 2 & 1 & -5 \\ 6 & 1 & -11 \\ -4 & -1 & 7 \end{pmatrix}$ 
```

```
(% o57) Ad =  $\begin{pmatrix} 2 & 6 & -4 \\ 1 & 1 & -1 \\ -5 & -11 & 7 \end{pmatrix}$ 
```

4.5. Matriz inversa

Para realizar la matriz inversa, podemos seleccionar la matriz y luego proceder en el menú **Álgebra** → **Invertir matriz**, o mediante el comando **invert**. Incluso valdría hacer A^{-1} , como vimos en las potencias de una matriz. (Véase 4.2.4)



```
(% i57) invert(A);
```

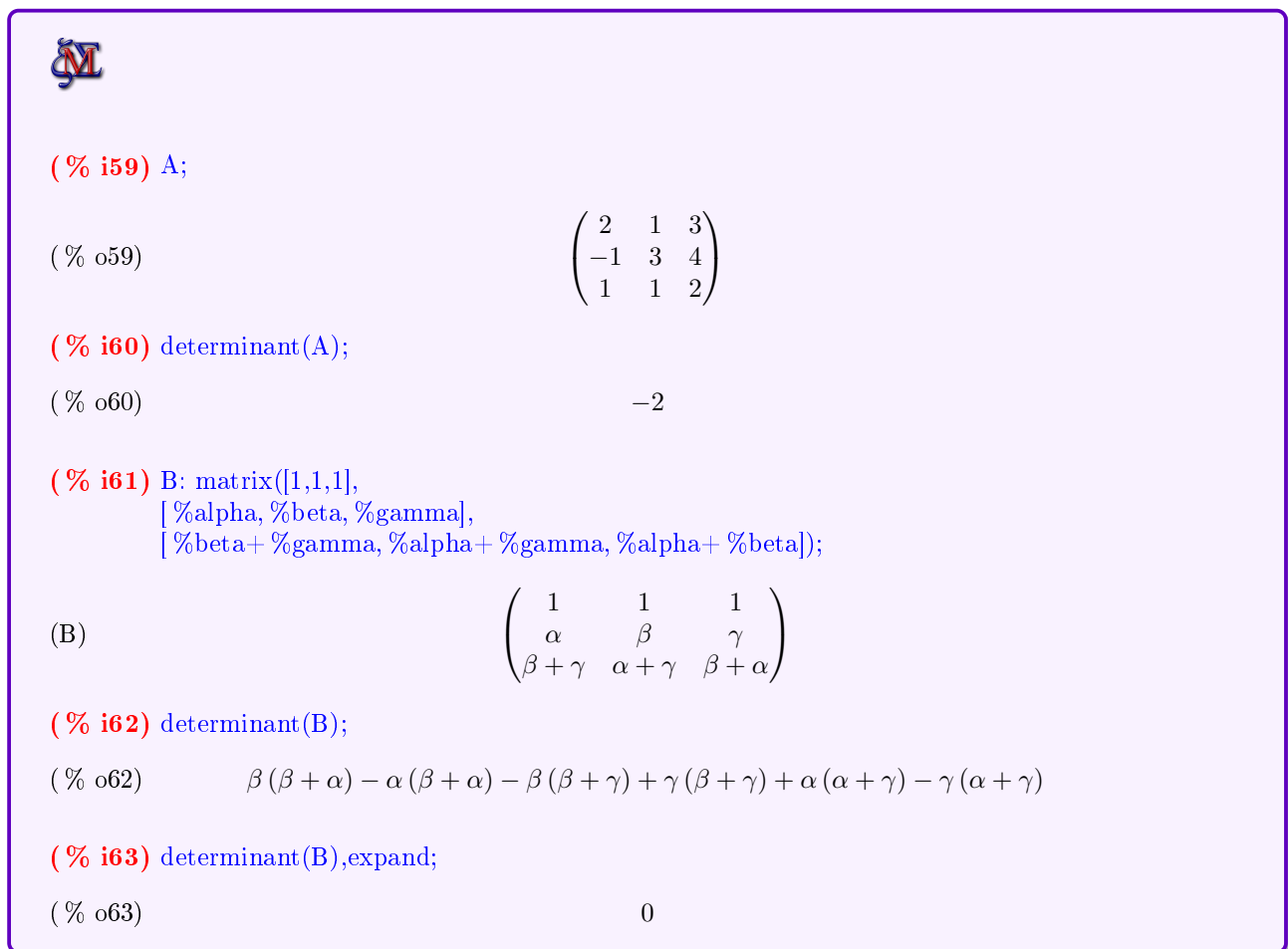
```
(% o57)  $\begin{pmatrix} -1 & -\frac{1}{2} & \frac{5}{2} \\ -3 & -\frac{1}{2} & \frac{11}{2} \\ 2 & \frac{1}{2} & -\frac{7}{2} \end{pmatrix}$ 
```

```
(% i58) A^^-1;
```

```
(% o58)  $\begin{pmatrix} -1 & -\frac{1}{2} & \frac{5}{2} \\ -3 & -\frac{1}{2} & \frac{11}{2} \\ 2 & \frac{1}{2} & -\frac{7}{2} \end{pmatrix}$ 
```

5. Determinante de una matriz

Podemos realizar el determinante de una matriz gráficamente si tras seleccionarla (cuando no es la última salida de Maxima) accedemos al menú **Álgebra** → **Determinante**. Se realiza mediante el comando `determinant`. Veamos algunos ejemplos:



The screenshot shows the Maxima CAS interface with the following content:

```
(% i59) A;
```

$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

```
(% i60) determinant(A);
```

(% o60) -2

```
(% i61) B: matrix([1,1,1],
  [%alpha,%beta,%gamma],
  [%beta+%gamma,%alpha+%gamma,%alpha+%beta]);
```

$$\begin{pmatrix} 1 & 1 & 1 \\ \alpha & \beta & \gamma \\ \beta + \gamma & \alpha + \gamma & \beta + \alpha \end{pmatrix}$$

```
(% i62) determinant(B);
```

(% o62) $\beta(\beta + \alpha) - \alpha(\beta + \alpha) - \beta(\beta + \gamma) + \gamma(\beta + \gamma) + \alpha(\alpha + \gamma) - \gamma(\alpha + \gamma)$

```
(% i63) determinant(B),expand;
```

(% o63) 0

Dejamos al lector la tarea de comprobar por qué $|B| = 0$ sin desarrollarlo.

6. Submatrices de una matriz

Podemos obtener una submatriz de otra matriz A eliminando alguna fila, columna o ambas cosas con el comando `submatrix`. Dicho comando tiene la siguiente sintaxis: `submatrix(fila, A, columna)`

Veamos algunos ejemplos:



(% i64) A;

(% o64)
$$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% i65) A1:submatrix(1,A) /* eliminamos la primera fila*/;

(A1)
$$\begin{pmatrix} -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% i66) A2:submatrix(A,2) /*eliminamos la segunda columna*/;

(A2)
$$\begin{pmatrix} 2 & 3 \\ -1 & 4 \\ 1 & 2 \end{pmatrix}$$

(% i67) A3:submatrix(1,A,2) /* eliminamos 1ª fila y 2ª columna*/;

(A3)
$$\begin{pmatrix} -1 & 4 \\ 1 & 2 \end{pmatrix}$$



Podemos así, por ejemplo, crearnos un comando para calcular el adjunto de un elemento a_{ij} .



(% i68) adjunto(A,i,j):=(-1)^(i+j)*determinant(submatrix(i,A,j)) /* eliminamos la fila i y la columna j */;

(% o68)
$$\text{adjunto}(A, i, j) := (-1)^{i+j} \text{determinant}(\text{submatrix}(i, A, j))$$

(% i69) adjunto(A,1,2) /* Calculamos el adjunto de a12*/;

(% o69)
$$6$$

(% i70) adjunta(A) /* Comprobamos que coincide en la matriz adjunta de A*/;

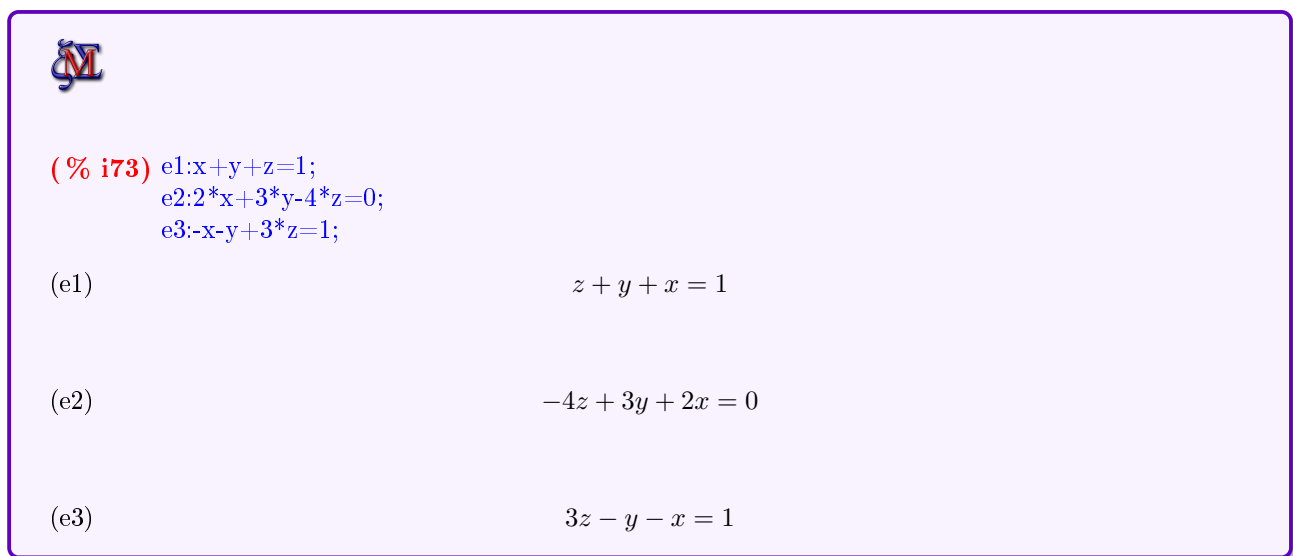
(% o70)
$$\begin{pmatrix} 2 & 6 & -4 \\ 1 & 1 & -1 \\ -5 & -11 & 7 \end{pmatrix}$$

7. Sistemas lineales con Maxima

En esta sección veremos la potencia de Maxima para resolver y discutir sistemas lineales. Veremos como hacerlo con el siguiente sistema de ejemplo:

$$\begin{cases} x + y + z = 1 \\ 2x + 3y - 4z = 0 \\ -x - y + 3z = 1 \end{cases}$$

Introducimos las ecuaciones del sistema, nombrándolas como e_1, e_2, e_3 respectivamente



```
(% i73) e1:x+y+z=1;
      e2:2*x+3*y-4*z=0;
      e3:-x-y+3*z=1;

(e1)                                z + y + x = 1

(e2)                                -4z + 3y + 2x = 0

(e3)                                3z - y - x = 1
```

Maxima cambia el orden de x, y, z

7.1. Resolver el sistema lineal

Si sólo queremos resolver el sistema, lo podemos hacer a través de los menús o mediante comandos. A través de menús, seleccionamos **Ecuaciones** → **Resolver sistema lineal**, que me abrirá un menú como el de la siguiente figura tras indicar el número de ecuaciones, en el que hemos introducido nuestras ecuaciones y las variables del sistema:

Obsérvese la sintaxis para resolver los sistemas lineales:

`linsolve([ecuaciones], [variables])`

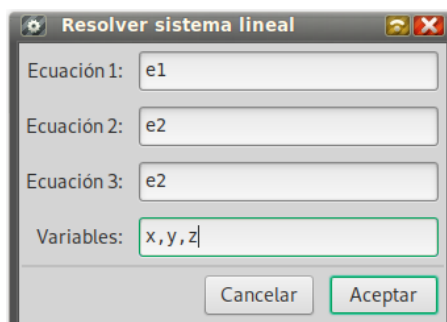


Figura 2: Menú para resolver un sistema lineal

Veamos el código y la solución obtenida:



```
(% i74) linsolve([e1, e2, e3], [x,y,z]);
```

```
(% o74) [x = -1/2, y = 1, z = 1/2]
```



¿Cómo resuelve Maxima un sistema compatible indeterminado?

Para dar respuesta a esa pregunta, crearemos otra ecuación $e_4 = e_1 + e_2$. El sistema formado por e_1, e_2, e_4 tendrá un grado de libertad. Procedamos:



```
(% i75) e4:e1+e2;
```

```
(e4) -3z + 4y + 3x = 1
```

```
(% i76) linsolve([e1,e2,e4],[x,y,z]);
```

solve: dependent equations eliminated: (3)

```
(% o76) [x = 3 - 7%r1, y = 6%r1 - 2, z = %r1]
```

☞ Maxima ha eliminado e_4 y ha tratado a z como parámetro.



Si queremos que el parámetro sea x , por ejemplo, se lo indicamos eliminándola de las variables:



```
(% i77) linsolve([e1,e2,e4],[y,z]);
```

solve: dependent equations eliminated: (3)

```
(% o77) [y = -6x - 4/7, z = -x - 3/7]
```

7.2. Matrices del sistema

Podemos crear las matrices de un sistema lineal directamente o bien a partir de las ecuaciones ya introducidas como mostraremos en los siguientes ejemplos, a partir del sistema formado por las ecuaciones e_1, e_2, e_3 vistas anteriormente:

☞ Con el comando `coefmatrix([ecuaciones],[variables])` obtenemos la matriz de los coeficientes:



```
(% i80) e1;  
      e2;  
      e3;
```

```
(% o78)          z + y + x = 1
```

```
(% o79)          -4z + 3y + 2x = 0
```

```
(% o80)          3z - y - x = 1
```

```
(% i81) A:coefmatrix([e1,e2,e3],[x,y,z]);
```

```
(A)          
$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & -4 \\ -1 & -1 & 3 \end{pmatrix}$$

```

☞ Creamos la matriz de los términos independientes. La manera más sencilla, es hacerlo gráficamente con una matriz de tres filas y una columna:



```
(% i82) B: matrix(  
      [1],  
      [0],  
      [1]  
      );
```

```
(B)          
$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

```

☞ Creamos la matriz ampliada con el comando `addcol`:



```
(% i83) Am:addcol(A,B);
```

```
(Am)           $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & -4 & 0 \\ -1 & -1 & 3 & 1 \end{pmatrix}$ 
```

7.3. Discusión de un sistema

7.3.1. Rango de una matriz

Para discutir un sistema, debemos comparar los rangos de la matriz de los coeficientes y de la ampliada. Para ello, utilizaremos el comando `rank(A)`:



```
(% i85) RA=rank(A);  
      RA=rank(Am);
```

```
(% o84)          RA = 3
```

```
(% o85)          RA = 3
```



Este sistema sería compatible determinado, ya que $R(A) = R(A') = 3 = n^{\circ}$ de incógnitas

7.4. Sistemas dependientes de algún parámetro

Entramos de lleno en el ejercicio típico de 2º de bachillerato, que consiste en discutir y resolver un sistema lineal dependiente de algún parámetro. Vamos a explicarlo a partir del siguiente sistema, que depende de los parámetros λ y μ :

☞ A partir del siguiente sistema:

$$\begin{cases} \lambda x + (\lambda + 1)y + z = 0 \\ -x + \lambda y - z = 0 \\ (\lambda - 1)x - y = 1 - \lambda \end{cases}$$

- Estudia la compatibilidad del siguiente sistema en función del parámetro λ
- Resuelve si $\lambda = 1$
- Resuelve por el método de la matriz inversa si $\lambda = 3$

☞ Introducimos las ecuaciones del sistema:



```
(% i88) e1:λ*x+(λ+1)*y+z=0;
      e2:-x+λ*y-z=0;
      e3:(λ-1)*x-y=1-λ;
```

$$(e1) \quad y(\lambda + 1) + x\lambda + z = 0$$

$$(e2) \quad y\lambda - z - x = 0$$

$$(e3) \quad x(\lambda - 1) - y = 1 - \lambda$$

☞ Creamos las matrices del sistema:



```
(% i89) A:coefmatrix([e1,e2,e3],[x,y,z]);
```

$$(A) \quad \begin{pmatrix} \lambda & \lambda + 1 & 1 \\ -1 & \lambda & -1 \\ \lambda - 1 & -1 & 0 \end{pmatrix}$$

```
(% i90) B: matrix(
      [0],
      [0],
      [1-λ]
      );
```

$$(B) \quad \begin{pmatrix} 0 \\ 0 \\ 1 - \lambda \end{pmatrix}$$

```
(% i91) Am:addcol(A,B);
```

$$(Am) \quad \begin{pmatrix} \lambda & \lambda + 1 & 1 & 0 \\ -1 & \lambda & -1 & 0 \\ \lambda - 1 & -1 & 0 & 1 - \lambda \end{pmatrix}$$

☞ Calculamos $|A|$ y obtenemos qué valores de λ lo hacen 0, obteniendo $\lambda = \pm 1$



```
(% i92) detA:determinant(A),expand;
```

```
(detA)                                2 - 2λ²
```

```
(% i93) solve(detA=0,λ);
```

```
(% o93)                                [λ = -1, λ = 1]
```



Si $\lambda \neq \pm 1 \Rightarrow R(A) = R(A') = 3 = n^{\text{o}} \text{ de incógnitas} \Rightarrow S.C.D$

☞ Vemos qué ocurre si $\lambda = -1$. Para ello, haremos uso del comando `at`, que utiliza la siguiente sintaxis:

`at(expresión, valores de las variables)`

Veamos cómo:



```
(% i94) A1:at(A,λ=-1);
```

```
(A1)                                ⎛ -1  0  1 ⎞  
                                ⎜ -1 -1 -1 ⎟  
                                ⎝ -2 -1  0 ⎠
```

```
(% i95) Am1:at(Am,λ=-1);
```

```
(Am1)                                ⎛ -1  0  1  0 ⎞  
                                ⎜ -1 -1 -1  0 ⎟  
                                ⎝ -2 -1  0  2 ⎠
```

☞ Calculamos los rangos de A_1 y Am_1 :



(% i96) rank(A1);

(% o96) 2

(% i97) rank(Am1);

(% o97) 3



$$R(A_1) = 2 < R(Am_1) = 3 \Rightarrow S.I.$$

☞ Vemos qué ocurre cuando $\lambda = 1$. Repetimos los pasos anteriores:



(% i98) A2:at(A,\lambda=1);

(A2)
$$\begin{pmatrix} 1 & 2 & 1 \\ -1 & 1 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

(% i99) Am2:at(Am,\lambda=1);

(Am2)
$$\begin{pmatrix} 1 & 2 & 1 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

(% i100) rank(A2);

(% o100) 2

(% i101) rank(Am2);

(% o101) 2



$$R(A_2) = R(Am_2) = 2 < n^{\circ} \text{ de incógnitas} \Rightarrow S.C.I \text{ con } 3 - 2 = 1 \text{ grado de libertad}$$

☞ Realizamos el apartado b), resolviendo cuando $\lambda = 1$. Para ello, vemos que las dos primeras ecuaciones son linealmente independientes, por lo que vamos a sustituir $\lambda = 1$ en ambas ecuaciones y a resolver:



(% i103) e12:at(e1,lambda=1);
e22:at(e2,lambda=1);

(e12) $z + 2y + x = 0$

(e22) $-z + y - x = 0$

(% i104) linsolve([e12,e22],[x,y,z]);

(% o104) $[x = -\%r2, y = 0, z = \%r2]$



La solución es

$$\begin{cases} x = -\mu \\ y = 0 \\ z = \mu \end{cases}$$

☞ Resolvemos el último apartado, obteniendo $A3$ y $B3$ como las matrices del sistema cuando $\lambda = 3$ y resolviendo con el método de la matriz inversa:

$$X = A^{-1} \cdot B$$



(% i105) A3:at(A,lambda=3);

(A3) $\begin{pmatrix} 3 & 4 & 1 \\ -1 & 3 & -1 \\ 2 & -1 & 0 \end{pmatrix}$

(% i106) B3:at(B,lambda=3);

(B3) $\begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix}$

(% i107) [x,y,z]=invert(A3).B3;

(% o107) $[x, y, z] = \begin{pmatrix} \frac{7}{8} \\ \frac{1}{4} \\ \frac{13}{8} \end{pmatrix}$

7.5. Triangularizar matrices: El método de Gauss

Podemos obtener la matriz triangular asociada a un sistema lineal, con el comando `triangularize(A)`, como se muestra a continuación:



```
(% i131) Am1;
```

```
(% o131) 
$$\begin{pmatrix} -1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ -2 & -1 & 0 & 2 \end{pmatrix}$$

```

```
(% i132) triangularize(Am1);
```

```
(% o132) 
$$\begin{pmatrix} -1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

```



Vemos que la última fila muestra $0 = 2 \Rightarrow S.I$

☞ Podemos incluso obtener la matriz triangular cuya diagonal principal esté formada por unos, con el comando `echelon(A)`:



```
(% i133) echelon(Am1);
```

```
(% o133) 
$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```

8. Inecuaciones

Maxima también resuelve inecuaciones. Para ello, hemos de cargar dos paquetes al inicio. Además, he definido dos funciones para hacer más sencilla la sintaxis de los comandos encargados de resolver dichas inecuaciones. Todo ello se puede ver al final de las ordenes definidas al al inicio, como se muestra en la sección 11.

8.1. Librerías y funciones *mías*



```
(% i4) load("fourier_elim")$ /* Para resolver inecuaciones lineales */
load(solve_rat_ineq); /* Para inecuaciones racionales*/
isolve(I,V):=fourier_elim(I,V); /* Función mía para resolver inecuaciones más sencilla en su sintaxis */
irsolve(I):=solve_rat_ineq(I); /* Ídem para inecuaciones racionales*/

(% o2)

"/usr/share/maxima/5.45.1/share/solve_rat_ineq/solve_rat_ineq.mac"

(% o3)          isolve(I, V) := fourier_elim(I, V)

(% o4)          irsolve(I) := solve_rat_ineq(I)
```

8.2. Inecuaciones de una variable

8.2.1. Inecuaciones de primer grado

Resolución de inecuaciones de una variable:

Inecuación de primer grado:

Resolvamos la siguiente inecuación:

$$\frac{5x - 2}{2} > 3x + 1$$



```
(% i5) ine1:(5*x-2)/2>3*x+1;

(ine1)          
$$\frac{5x - 2}{2} > 3x + 1$$


(% i6) isolve(ine1,[x])/* Utilizo "mi función"*/;

(% o6)          [x < -4]
```

8.2.2. Inecuaciones de 2º grado

Inecuaciones de segundo grado:

Resolvamos la siguiente inecuación:

$$x^2 + 5x - 6 \geq 0$$



```
(% i7) ine2:x^2+5*x-6>=0;
```

```
(ine2)
```

$$x^2 + 5x - 6 \geq 0$$

```
(% i8) solve(ine2,[x]);/* Nótese el resultado cuando se permite la igualdad en la inecuación*/
```

```
(% o8)
```

$$[x = -6] \text{ or } [x = 1] \text{ or } [1 < x] \text{ or } [x < -6]$$

8.2.3. Inecuaciones polinómicas

También podemos resolverlas con nuestra función *isolve*:

Ejemplo:

$$x^4 - 3x^3 - 3x^2 + 11x - 6 > 0$$



```
(% i9) ine3:(x-1)^2*(x+2)*(x-3)>0,expand;
```

```
(ine3)
```

$$x^4 - 3x^3 - 3x^2 + 11x - 6 > 0$$

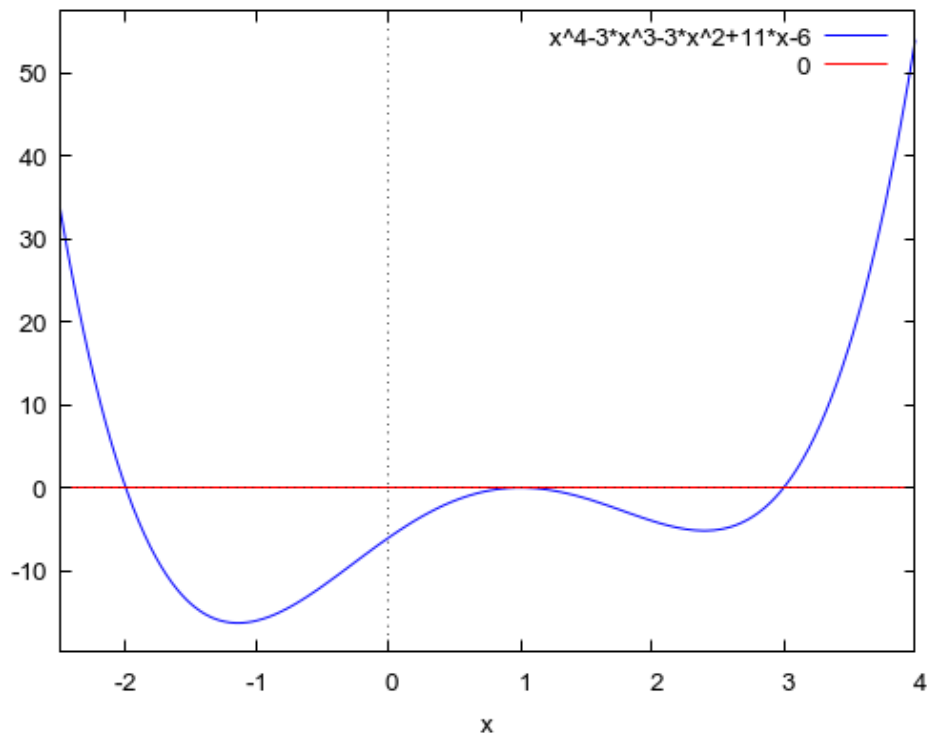
```
(% i10) isolve(ine3,[x]);
```

```
(% o10)
```

$$[3 < x] \text{ or } [x < -2]$$

Comprobación gráfica:

```
(% i11) wxplot2d([x^4-3*x^3-3*x^2+11*x-6,0], [x,-2.5,4])$
(% t11)
```



8.2.4. Inecuaciones con valores absolutos

Veamos un ejemplo:

$$\left| \frac{x-8}{x-1} \right| \geq 1$$



```
(% i13) ine4:abs((x-8)/(x-1))>=1;
isolve(ine4,[x]);
```

(ine4) $\left| \frac{x-8}{x-1} \right| \geq 1$

(% o13) $\left[x = \frac{9}{2} \right]$ or $[x < 1]$ or $[x = 1]$ or $\left[1 < x, x < \frac{9}{2} \right]$

8.2.5. Inecuaciones racionales

Para resolverlas, utilizo mi *segunda función*:

$$\frac{1}{x} < x^2$$



```
(% i14) ine1:1/(x)<x^ 2;
```

```
(ine1) 
$$\frac{1}{x} < x^2$$

```

```
(% i15) irsolve(ine1)/ * No es necesario indicar la variable */;
```

```
(% o15) 
$$[[x < 0], [x > 1]]$$

```

Intento de resolución como inecuación "normal":

```
(% i16) isolve(ine1,[x]);/ * Al ser una inecuación racional, no da la solución */
```

```
(% o16) 
$$[1 < x, x^2 + x + 1 > 0] \text{ or } [x < 0, x^2 + x + 1 > 0] \text{ or } [0 < x, x < 1, -(x^2 + x + 1) > 0]$$

```

8.3. Sistemas de inecuaciones de una variable

Resolvamos el siguiente sistema como ejemplo:

$$\begin{cases} 2x - 3 < 5x + 2 \\ x - 5 < 8 \end{cases}$$



```
(% i18) in1:2*x-3<5*x+2;  
in2:x-5<8;
```

```
(in1) 
$$2x - 3 < 5x + 2$$

```

```
(in2) 
$$x - 5 < 8$$

```

```
(% i19) isolve(in1 and in2,[x]);
```

```
(% o19) 
$$\left[-\frac{5}{3} < x, x < 13\right]$$

```

8.4. Inecuaciones de dos variables

Vamos a ver cómo despejar y en la siguiente inecuación:

$$y + x > 5$$



```
(% i21) in:x+y>5;  
       isolve(in,[y]);
```

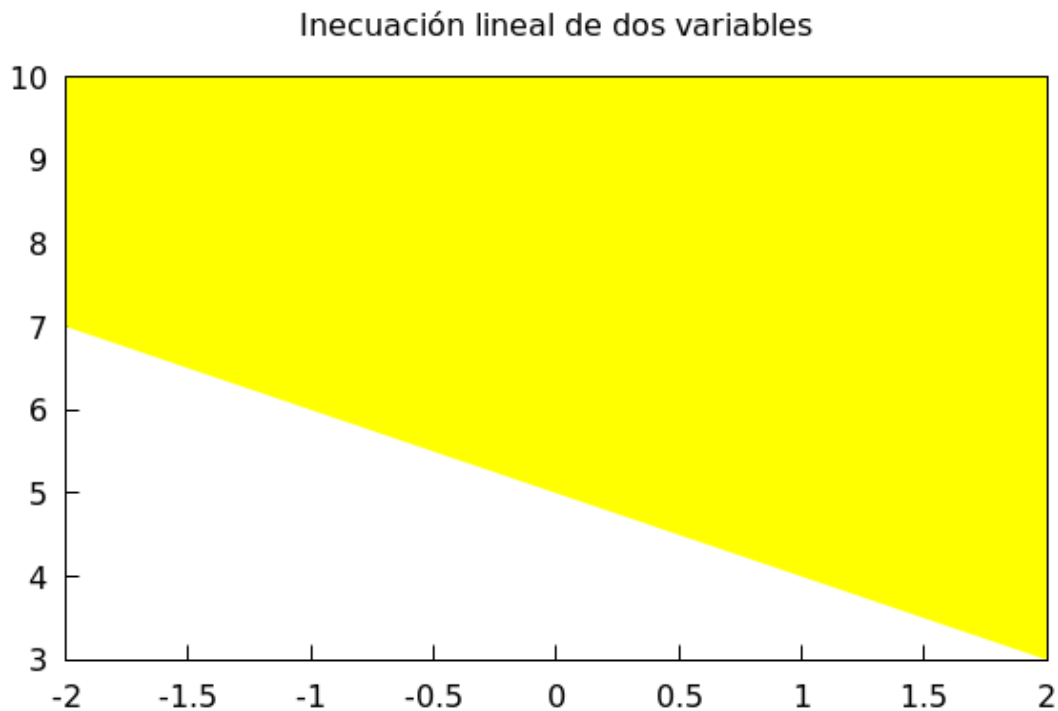
```
(in)                                     y + x > 5
```

```
(% o21)                                  [5 - x < y]
```

Podemos representar la solución anterior rellenando el área con la función $y = 10$, por ejemplo, ya que y ha quedado en el *lado del mayor* en la desigualdad:



```
(% i22) wxdraw2d(
  filled_func=10,
  fill_color="≠ffff00",
  explicit(
    5-x,
    x,-2,2
  ),
  title="Inecuación lineal de dos variables"
)$
(% t22)
```



8.5. Recintos

Vamos a ver cómo representar recintos definidos por inecuaciones lineales, muy utilizados en la programación lineal de 2º de bachillerato. Veamos un par de ejemplos. En este primero, a la inecuación $y < 3 - 2x$ le añadiremos la restricción de que las variables han de ser positivas:

$$\begin{cases} y < 3 - 2x \\ x > 0 \\ y > 0 \end{cases}$$

Veamos cómo resolverlo y representarlo:



(% i23) in2:y<3-2*x;

(in2) $y < 3 - 2x$

(% i24) isolve(in2 and x>0 and y>0,[x,y]);

(% o24) $\left[0 < x, x < \frac{3}{2} - \frac{y}{2}, 0 < y, y < 3 \right]$

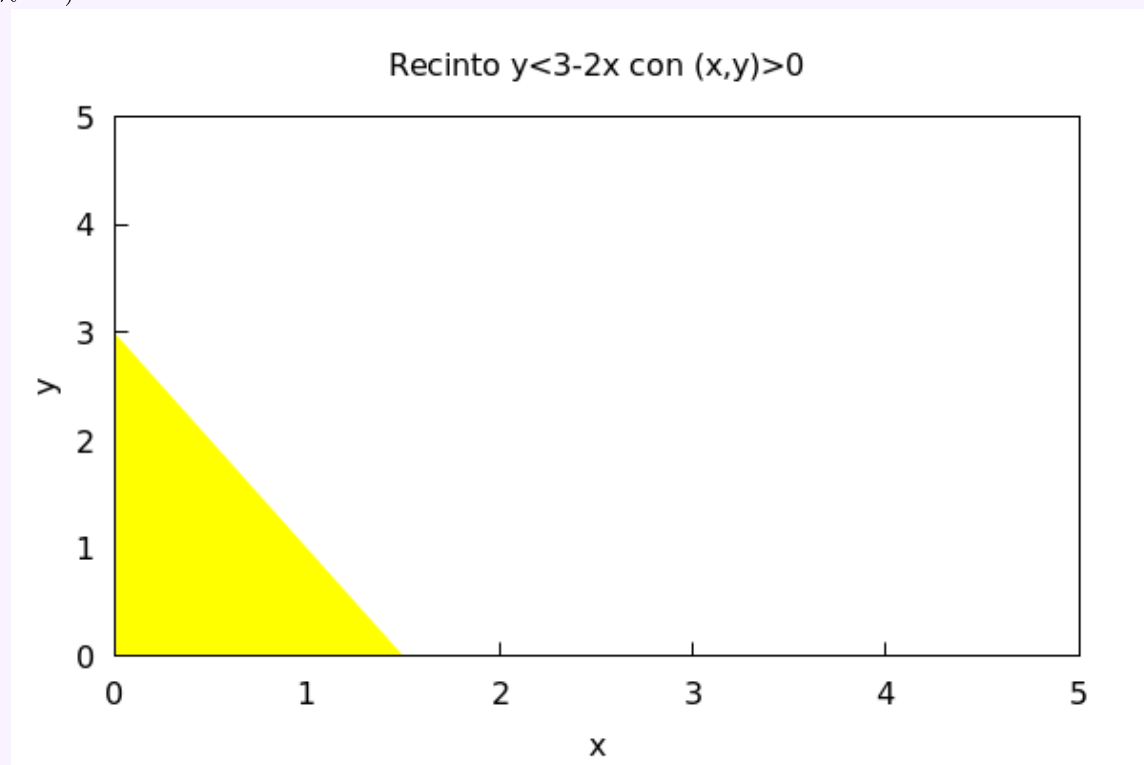
(% i25) isolve(%[2],[y]) /* Despejamos y */;

(% o25) $[y < 3 - 2x]$

Hemos obtenido como solución que $y < 3 - 2x$, con $x > 0$ e $y \in (0, 3)$. Vamos a representar dicho recinto:



```
(% i26) wxdraw2d(  
  filled_func=0,  
  fill_color="ffff00",  
  explicit(  
    3-2*x,  
    x,0,3/2  
  ),  
  xlabel="x",  
  ylabel="y",  
  xrange=[0,5],  
  yrange=[0,5],  
  title="Recinto  $y < 3 - 2x$  con  $(x,y) > 0$ "  
)$  
(% t26)
```



Otro ejemplo:

$$\begin{cases} x + y < 5 \\ 2x + y > 3 \\ x > 0 \\ y > 0 \end{cases}$$



```
(% i28) in1:x+y<5;
      in2:2*x+y>3;
```

```
(in1)                                     y + x < 5
```

```
(in2)                                     y + 2x > 3
```

```
(% i29) R:solve(in1 and in2 and [x,y]>[0,0],[x,y]);
```

```
(R)                                     [ 3/2 - y/2 < x, x < 5 - y, y < 7, [x, y] > 0 ]
```

```
(% i30) isolve(R,[y]);/* con y despejada*/
```

```
(% o30) [3 - 2x < y, y < 5 - x, 2(x + 2) > 0, x + 2 > 0, [x, y] > 0]
```

Representamos el recinto:

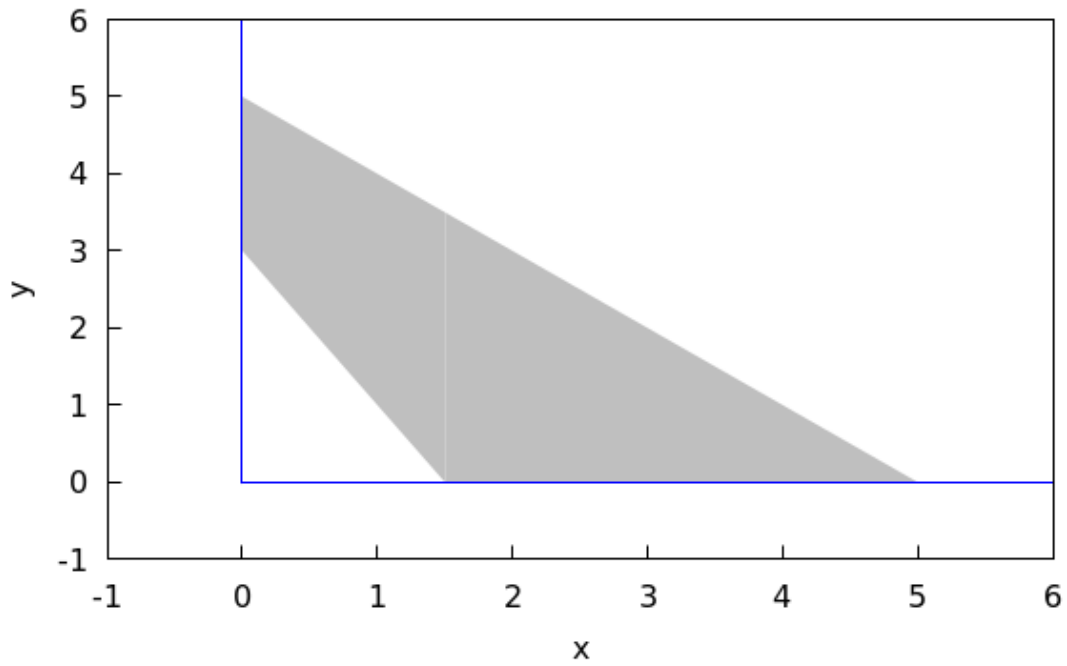


```
(% i36) wxdraw2d(
  filled_func=5-x,
  fill_color="≠bfbfbf",
  explicit(
    3-2*x,
    x,0,3/2
  ),explicit(0,x,3/2,5),
  parametric(
    0,
    t,
    t,0,6
  ),
  parametric(
    t,
    0,
    t,0,6
  ),
  filled_func=false,
  title="Ejemplo de recinto (II)",
  xlabel="x",
  ylabel="y",
  xrange=[-1,6],
  yrange=[-1,6])$
```



(% t36)

Ejemplo de recinto (II)



9. Vectores

9.1. Cargar una librería necesaria



Para trabajar con vectores, hemos de cargar la librería `vect`, para tener definidos algunos de los comandos a utilizar. Para ello, cargamos dicha librería:



```
(% i108) load(vect);
vect: warning: removing existing rule or rules for ".".

(% o108)

/usr/share/maxima/5.42.1/share/vector/vect.mac
```

9.2. Funciones propias útiles para vectores

Es muy útil definir tres funciones, una para calcular el producto vectorial, otra para calcular el módulo de un vector y una tercera para realizar el producto mixto de tres vectores. Las podemos definir así, por ejemplo:



```
(% i111) vect(u,v):=express(u~v);
          modulo(u):=sqrt(u.u);
          pmixto(u,v,w):=determinant(matrix(u,v,w));

(% o109)                                vect(u,v) := express(u ~ v)

(% o110)                                modulo(u) := sqrt(u.u)

(% o111)                                pmixto(u,v,w) := determinant(matrix(u,v,w))
```

9.3. Introducción de vectores

☞ Veamos cómo podemos introducir los vectores $\vec{u} = (1, 2, 3)$ y $v = (-1, 1, 2)$



(% i113) u:[1,2,3];
v:[-1,-1,2];

(u) [1, 2, 3]

(v) [-1, -1, 2]

9.4. Operaciones básicas con vectores

☞ Veamos cómo realizar $\vec{w} = 2\vec{u} + 3\vec{v}$



(% i114) w:2*u+3*v;

(w) [-1, 1, 12]

9.5. Producto escalar



Hemos de utilizar el punto *normal*, al igual que en el producto de matrices. (Véase la sección 4.2.3)

☞ Realicemos $\vec{u} \cdot \vec{v}$



(% i115) u.v;

(% o115) 3

9.6. Módulo de un vector



Hemos definido anteriormente la función módulo a partir de la siguiente propiedad:

$$|\vec{u}| = \sqrt{\vec{u} \cdot \vec{u}}$$

Así, utilizamos esa función para calcular el módulo de \vec{u} y \vec{v} :



```
(% i116) modulo(u):=sqrt(u.u);
```

```
(% o116)                                modulo(u) :=  $\sqrt{u \cdot u}$ 
```

```
(% i117) modulo(u);
```

```
(% o117)                                 $\sqrt{14}$ 
```

```
(% i118) modulo(v);
```

```
(% o118)                                 $\sqrt{6}$ 
```

9.7. Vectores unitarios



Recuerda: $\hat{u} = \frac{\vec{u}}{|\vec{u}|}$ Vamos a calcular los vectores unitarios de \vec{u} y \vec{v} , comprobando que su módulo es 1:



```
(% i120) uu:u/modulo(u);  
          vu:v/modulo(v);
```

```
(uu)                                 $[\frac{1}{\sqrt{14}}, \frac{2}{\sqrt{14}}, \frac{3}{\sqrt{14}}]$ 
```

```
(vu)                                 $[-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}]$ 
```

```
(% i122) modulo(uu);  
          modulo(vu);
```

```
(% o121)                                1
```

```
(% o122)                                1
```

9.8. Producto vectorial

He definido una función para el producto vectorial, ya que su sintaxis no es sencilla de recordar:



(% i123) vect(u,v):=express(u~v);

(% o123) vect(u,v) := express(u ~ v)

(% i124) w:vect(u,v);

(w) [7, -5, 1]



Comprobamos que $\vec{w} \perp \vec{u}$ y $\vec{w} \perp \vec{v}$, siendo esta una propiedad fundamental del producto vectorial:



(% i126) u.w;
v.w;

(% o125) 0

(% o126) 0

9.9. Producto mixto



Aplicamos la regla para el cálculo de producto mixto de tres vectores:

$$[\vec{u}, \vec{v}, \vec{w}] = \vec{u} \cdot (\vec{v} \times \vec{w}) = \begin{vmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{vmatrix}$$

Definimos una función que calcule el determinante de la matriz formada por los tres vectores:



(% i127) pmixto(u,v,w):=determinant(matrix(u,v,w));

(% o127) pmixto(u,v,w) := determinant(matrix(u,v,w))

(% i128) pmixto(u,v,w);

(% o128) 75

10. Geometría espacial con Maxima

En esta sección vamos a ver como trabajar los principales conceptos de la geometría de 2º de Bachillerato. Haremos uso de funciones definidas al arrancar Maxima, como se puede ver en la sección 11:

10.1. Rectas en el espacio

Podemos introducir rectas a partir de sus elementos o de cualquiera de sus ecuaciones:



Introducción de una recta a partir de sus elementos

Dados u y Pr :

(% i2) $u:[1,2,3];Pr:[-1,1,2];$

(u) $[1, 2, 3]$

(Pr) $[-1, 1, 2]$

Ecuación vectorial y ecuaciones paramétricas Podemos introducir dichas ecuaciones a partir de los elementos anteriores. Incluso, podemos crear una función *parametricas* para obtener dichas ecuaciones a partir de P_r y \vec{u}



Ecuación vectorial:

(% i3) $r:Pr+\lambda*u;$

(r) $[\lambda - 1, 2\lambda + 1, 3\lambda + 2]$

(% i4) $parametricas(Pr,u):=Pr+\lambda*u$

(% i5) $r:parametricas(Pr,u);$

(r) $[\lambda - 1, 2\lambda + 1, 3\lambda + 2]$

Introducción de una recta r dadas sus ecuaciones implícitas: ¿Cómo introducimos las ecuaciones de esta recta?

$$r \equiv \begin{cases} x + y + z = -1 \\ -2x + 3y + 4z = 2 \end{cases}$$

Podemos introducirla como un *vector* de dos coordenadas, con las ecuaciones de r . Veamos cómo:



Recta r a partir de sus ecuaciones implícitas:

(% i6) `r:[x+y+z=-1,-2*x+3*y+4*z=2];`

(r) $[z + y + x = -1, 4z + 3y - 2x = 2]$

Obtención de sus ecuaciones paramétricas a partir de sus ecuaciones implícitas Basta con resolver el sistema, y asignar al parámetro de Maxima λ , para que *quede bonito*:



Obtención de las coordenadas paramétricas:

(% i8) `var:[x,y,z];/* Variables de las ecuaciones*/
rp:linsolve(r,var);`

(var) $[x, y, z]$

(rp) $\left[x = \frac{\%r1 - 5}{5}, y = -\frac{6\%r1}{5}, z = \%r1 \right]$

Cambiamos $\%r1$ por λ :

(% i9) `rp:at(rp,%r1=λ);`

(rp) $\left[x = \frac{\lambda - 5}{5}, y = -\frac{6\lambda}{5}, z = \lambda \right]$

A partir de aquí, podríamos definir P_r y \vec{u}_r , etc...

Ecuación continua De nuevo las podemos introducir con un *vector de ecuaciones*. Veamos un ejemplo:

$$r \equiv \frac{x-3}{2} = \frac{y+5}{3} = z-2$$



Ecuación continua:

(% i10) `r:[(x-3)/2=(y+5)/3,(y+5)/3=z-2];`

(r) $\left[\frac{x-3}{2} = \frac{y+5}{3}, \frac{y+5}{3} = z-2 \right]$

Obtenemos las paramétricas de nuevo resolviendo el sistema:

(% i11) `rp:linsolve(r,var);`

(rp) $[x = 2\%r2 - 1, y = 3\%r2 - 11, z = \%r2]$

Puesto bonito:

(% i12) rp:at(rp,%r2=λ);

(rp) $[x = 2\lambda - 1, y = 3\lambda - 11, z = \lambda]$

10.2. Planos

Podemos introducir planos, bien a través de su ecuación general, elementos, etc... Lo más sencillo si se dispone de la ecuación general es introduciendo su vector normal y el término d de la ecuación general $ax + by + cz + d = 0$. Veamos un ejemplo:

Dado el plano $\pi \equiv 2x - 3y + 6z - 8 = 0$, obtener su vector normal y sus ecuaciones paramétricas:



Planos

$r \equiv 2x - 3y + 6z - 8 = 0$

(% i14) n:[2,-3,6]; /* Vector normal*/
d:-8;

(n) $[2, -3, 6]$

(d) -8

(% i15) pi:n.var+d=0;

(pi) $6z - 3y + 2x - 8 = 0$

Coordenadas paramétricas resolviendo el sistema de su ecuación:

(% i16) Ppi:linsolve(pi,var);

(Ppi) $\left[x = -\frac{-3\%r4 + 6\%r3 - 8}{2}, y = \%r4, z = \%r3 \right]$

Bonito:

(% i17) Ppi:at(Ppi,[%r4=λ,%r3=μ]);

(Ppi) $\left[x = -\frac{6\mu - 3\lambda - 8}{2}, y = \lambda, z = \mu \right]$

Problema inverso: ecuación general a partir de sus ecuaciones paramétricas: Veamos otro ejemplo:

Dado el plano $\pi \equiv \begin{cases} x = 2 + \lambda - \mu \\ y = 5 + 2\lambda - 3\mu \\ z = 1 + 2\mu \end{cases}$, determine su ecuación general:¹

¹Utilizaré el producto vectorial definido en la sección 9.8



Ecuación general desde sus ecuaciones paramétricas:

(% i18) pi:[2+λ-μ,5+2*λ-3*μ,1+2*μ];

(pi) $[-\mu + \lambda + 2, -3\mu + 2\lambda + 5, 2\mu + 1]$

Obtenemos un punto del plano y dos vectores u y v contenidos en el mismo:

(% i19) P:at(pi,[λ=0,μ=0]);

(P) $[2, 5, 1]$

(% i21) u:[1,2,0];
v:[-1,-3,2];

(u) $[1, 2, 0]$

(v) $[-1, -3, 2]$

Vector normal:

(% i22) n:vect(u,v);

(n) $[4, -2, -1]$

Hallamos d imponiendo la pertenencia de P a pi. Para ello antes libero el valor de d:

(% i23) remvalue(d);

(% o23) $[d]$

(% i24) solve(n.P+d=0,d);

(% o24) $[d = 3]$

Obtengo pi:

(% i25) pi:n.var+3=0;

(pi) $-z - 2y + 4x + 3 = 0$

10.3. Intersecciones

Intersecciones entre los distintos elementos:

10.3.1. Intersección entre dos rectas

Veamos algunos ejemplos:

Determine el punto de corte entre las rectas r y s :

$$r : \begin{cases} x + y - z = 6 \\ x + z = 3 \end{cases} \quad s : \frac{x-1}{-1} = \frac{y+1}{6} = \frac{z}{2}$$



Intersección entre dos rectas:

(% i27) $r: [x+y-z=6, x+z=3];$
 $s: [(x-1)/(-1)=(y+1)/6, (y+1)/6=z/2];$

(r) $[-z + y + x = 6, z + x = 3]$

(s) $\left[1 - x = \frac{y+1}{6}, \frac{y+1}{6} = \frac{z}{2} \right]$

Resolviendo el sistema:

(% i28) $Pc:linsolve([r[1],r[2],s[1],s[2]],var);$

solve: dependent equations eliminated: (4)

(Pc) $[x = -1, y = 11, z = 4]$

Segundo ejemplo: Las rectas son las mismas, pero r la tenemos *en paramétricas*:



Mismo ejemplo pero con r en paramétricas:

(% i29) $Pr:linsolve(r,var);$

(Pr) $[x = 3 - \%r5, y = 2 \%r5 + 3, z = \%r5]$

(% i30) $Pr:at(Pr, \%r5=\lambda)/* Usamos \lambda como parámetro */;$

(Pr) $[x = 3 - \lambda, y = 2\lambda + 3, z = \lambda]$

Sustituimos en las ecuaciones de s

(% i31) $E:at(s,Pr);$

(E) $\left[\lambda - 2 = \frac{2\lambda + 4}{6}, \frac{2\lambda + 4}{6} = \frac{\lambda}{2} \right]$

(% i32) $\lambda1:solve(E[1]); /* Resolvemos la primera de las ecuaciones, por ejemplo*/$

(\lambda1) $[\lambda = 4]$

Obtenemos el punto de corte:

(% i33) $Pc:at(Pr,\lambda1);$

(Pc) $[x = -1, y = 11, z = 4]$

10.3.2. Intersección recta-plano

Ejemplo de intersección de una recta y un plano En este ejemplo vemos cómo calcular la intersección de una recta r , de la que conocemos su vector de dirección y un punto, con un plano π expresado con su ecuación general.²



Intersección recta plano:

Si la recta está expresada con sus ecuaciones implícitas, resolvemos el sistema. Veamos un ejemplo con recta en paramétricas:

```
(% i36) Pr:[1,-1,1];
        ur:[2,1,3];
        r:parametricas(Pr,ur);
```

```
(Pr)                                     [1, -1, 1]
```

```
(ur)                                     [2, 1, 3]
```

```
(r)                                     [2λ + 1, λ - 1, 3λ + 1]
```

```
(% i37) pi:2*x+3*y-2*z+8=0;
```

```
(pi)                                     -2z + 3y + 2x + 8 = 0
```

Elementos de pi:

```
(% i39) n:[2,3,-2];
        d:8;
```

```
(n)                                     [2, 3, -2]
```

```
(d)                                     8
```

Sustituimos las ecuaciones paramétricas en pi:

```
(% i40) e1:n.r+d=0;
```

```
(e1)                                     -2(3λ + 1) + 2(2λ + 1) + 3(λ - 1) + 8 = 0
```

```
(% i41) e1,expand/* Agrupamos términos*/;
```

```
(% o41)                                     λ + 5 = 0
```

```
(% i42) λ1:solve(e1);
```

```
(λ1)                                     [λ = -5]
```

Hallamos el punto de corte:

```
(% i43) Pc:at(r,λ1);
```

```
(Pc)                                     [-9, -6, -14]
```

²Si la recta estuviera expresada con sus ecuaciones implícitas o la continua, bastaría con resolver el sistema formado por todas las ecuaciones.

10.3.3. Intersección entre dos planos

Para realizar su intersección, si estos están dados a partir de su ecuación general, resolvemos sus sistema, *ponemos bonito* el parámetro dado y damos la recta pedida (si es que se cortan). Si alguno está en forma paramétrica, lo más sencillo es obtener su ecuación general como vimos en 10.2.



Intersección entre planos

(% i45) pi:2*x+3*y-4*z+2=0;
sigma:-3*x+y-3*z+1=0;

$$(pi) \quad -4z + 3y + 2x + 2 = 0$$

$$(\sigma) \quad -3z + y - 3x + 1 = 0$$

Resolvemos el sistema:

(% i46) ri:linsolve([pi,sigma],var);

$$(ri) \quad \left[x = -\frac{5\%r6 - 1}{11}, y = \frac{18\%r6 - 8}{11}, z = \%r6 \right]$$

(% i47) ri:at(ri, %r6=λ);

$$(ri) \quad \left[x = -\frac{5\lambda - 1}{11}, y = \frac{18\lambda - 8}{11}, z = \lambda \right]$$

Si queremos que por ejemplo y sea λ, podemos hacerlo de la siguiente manera

(% i48) ry:linsolve([pi,sigma],[x,z]);

$$(ry) \quad \left[x = -\frac{5y + 2}{18}, z = \frac{11y + 8}{18} \right]$$

(% i49) ry:at(ry,y=λ);

$$(ry) \quad \left[x = -\frac{5\lambda + 2}{18}, z = \frac{11\lambda + 8}{18} \right]$$

Escribimos la solución en forma vectorial, por ejemplo:

(% i50) ri:var=[ry[1],y=λ,ry[2]];

$$(ri) \quad [x, y, z] = \left[x = -\frac{5\lambda + 2}{18}, y = \lambda, z = \frac{11\lambda + 8}{18} \right]$$

10.4. Distancias

En esta sección veremos como calcular la distancia entre los distintos elementos:

10.4.1. Distancia entre dos puntos

Para calcular la distancia entre dos puntos, basta con calcular el módulo del vector que los une. Para ello, utilizaré una función mía definida al inicio, `modulo(u)`, como puede verse en la sección 11.



Distancia entre dos puntos P y Q:

(% i52) P:[1,2,3];
Q:[-1,1,2];

(P) [1, 2, 3]

(Q) [-1, 1, 2]

Vector PQ

(% i53) PQ:Q-P;

(PQ) [-2, -1, -1]

Distancia d:

(% i54) d:modulo(PQ);

(d) $\sqrt{6}$

10.4.2. Distancia de un punto a una recta

Para calcularla haremos uso de la siguiente relación:

$$d(P, r) = \frac{|\overrightarrow{P_r P} \times \vec{u}_r|}{|\vec{u}_r|}$$

Veamos un ejemplo:

Calcula la distancia de un punto $P = (1, 2, 3)$ a la recta $r \equiv \frac{x-2}{3} = \frac{y-1}{-1} = \frac{z+4}{2}$.



Distancia de un punto a una recta

(% i56) P:[1,2,3];
r:[(x-2)/3=(y-1)/(-1),(x-2)/3=(z+4)/2];

(P) [1, 2, 3]

(r) $\left[\frac{x-2}{3} = 1-y, \frac{x-2}{3} = \frac{z+4}{2} \right]$

Elementos de r:

(% i58) Pr:[2,1,-4];
ur:[3,-1,2];

(Pr) [2, 1, -4]

(ur) [3, -1, 2]

Vector PrP

(% i59) PrP:P-Pr;

(PrP) [-1, 1, 7]

Distancia r,P

(% i60) d:modulo(vect(ur,PrP))/modulo(ur);

(d) $\frac{\sqrt{614}}{\sqrt{14}}$

10.4.3. Distancia de un punto a un plano

Para calcularlo, haremos uso de la siguiente relación:

$$d(P, \pi) = \frac{|a \cdot P_x + b \cdot P_y + c \cdot P_z + d|}{|\vec{n}|}$$

Veamos el siguiente ejemplo:

Calcúlese la distancia del punto $P = (1, 2, -1)$ al plano $\pi \equiv 2x - 3y + 4z = 5$



Distancia de un punto a un plano:

(% i62) P:[1,2,-1];
pi:2*x-3*y+4*z=5;

(P) [1, 2, -1]

(pi) $4z - 3y + 2x = 5$

(% i64) n:[2,-3,4];
d:-5;

(n) [2, -3, 4]

(d) -5

(% i65) D:abs((n.P+d)/modulo(n));

(D)

$$\frac{13}{\sqrt{29}}$$

11. Configurar arranque de Maxima

En nuestra carpeta personal, hay una carpeta oculta con el nombre `.maxima`³, donde existe un archivo de configuración llamado `maxima-init.mac`. En dicho archivo, podemos escribir todas las órdenes que queremos que se ejecuten al abrir Maxima. Sirva como ejemplo el mío, que recoge muchas de las órdenes personalizadas de este manual:

Mi archivo `maxima-init.mac`

```

logd(x):=log(x)/log(10);/* Para logaritmos decimal*/
loga(x,b):=log(x)/log(b);/* Cambio de base*/
Mt:5.97e24;/* Masa de la tierra*/
G:6.67e-11;/* Constante G*/
Rt:6.37e6;/* Radio terrestre*/
rad(x):=%pi*x/180;/* Conversión a radianes*/
gra(x):=180*x/%pi;/* Conversión a grados*/
load(vect);/* Carga paquete vectores*/
vect(u,v):=express(u~v);/* Producto vectorial*/
modulo(u):=sqrt(u.u);/* Cálculo de |u|*/
pmixto(u,v,w):=determinant(matrix(u,v,w));/* producto mixto*/
Io:1e-12;/* Intensidad umbral*/
Ampliada(E,V):=addcol(submatrix(augcoefmatrix(E,V),matrix_size(augcoefmatrix(E,V))[2]),
col(augcoefmatrix(E,V),matrix_size(augcoefmatrix(E,V))[2])*(-1));/* Genera la matriz ampliada: E son
las ecuaciones entre corchetes y V las variables, también entre corchetes*/
load("fourier_elim")$ /* Para realizar inecuaciones */
isolve(I,V):=fourier_elim(I,V); /* Función mía para resolver inecuaciones más sencilla en su sintaxis */
load(solve_rat_ineq); /* Para inecuaciones racionales*/
irsolve(I):=solve_rat_ineq(I); /* Ídem para inecuaciones racionales*/
    
```

³Podemos ver en qué carpeta se halla ejecutando `maxima_userdir` en WxMaxima