WxMaxima para Bachillerato

David Matellano

2 de mayo de 2020

I.E.S. Ángel Corella

Departamento de Matemáticas

Índice

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:

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

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

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

(% o22) $\text{loga}(x,a) := \frac{\log(x)}{\log(a)}$

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) $$\mathrm{gra}(x) := \frac{180x}{\pi}$$

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

(% o25) $$60$$

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

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

(% 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, $M_t = 5,97 \cdot 10^{24}$ *kg; constante de gravitación universal,* $G = 6,67 \cdot 10^{-11}$ Nm^2/kg^2; radio terrestre, $R_t = 6370$ km

Recuerda la expresión de $|\vec{g}|$:

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

☞Introducimos los datos y operamos

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

(Mt) $5{,}97 10^{24}$

(G) $6{,}67 10^{-11}$

(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 $\hat{i} = 30°$

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

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

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

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:

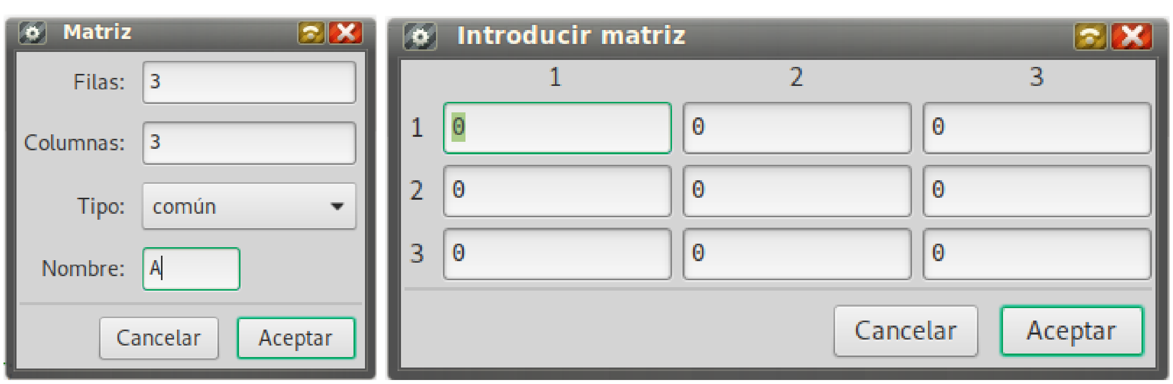
```
(%i34) n1:1.5;
       n2:1.33;
       i:30 /* grados */;
(n1)                    1,5
(n2)                    1,33
(i)                     30
(%i35) r:gra(asin(n2*sin(rad(i))/n1)),numer;
(r)                     26,31675512821711
```

4. Matrices y determinantes

Vamos a ver que esta es una de las herramientas más potentes de Maxima para el estudiante de 2.º 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 Álbegra→ 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 ˆ **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=Aˆ2;Aˆˆ2;

(% 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 trasponer 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 trasponer 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) $$\mathrm{adjunta}(A) := \mathrm{transpose}\,(\mathrm{adjoint}(A))$$

(% 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 (cuandoo no es la última salida de Maxima) accedemos al menú Álgebra→ Determinante. Se realiza mediante el comando determinant. Veamos algunos ejemplos:

(% i59) A;

(% o59) $$\begin{pmatrix} 2 & 1 & 3 \\ -1 & 3 & 4 \\ 1 & 1 & 2 \end{pmatrix}$$

(% i60) determinant(A);

(% o60) $$-2$$

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

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

(% 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:

$\boxed{submatrix(fila, A, columna)}$ 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:

$$\boxed{linsolve([ecuaciones],[variables])}$$

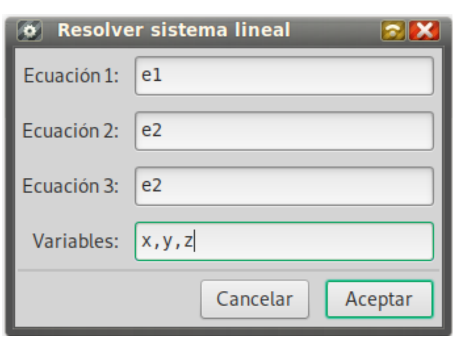


Figura 2: Menú para resolver un sistema lineal

Veamos el código y la solución obtenida:

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

(% o74) $$[x=-\frac{1}{2}, y=1, z=\frac{1}{2}]$$

¿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) $$[y=-\frac{6x-4}{7}, z=-\frac{x-3}{7}]$$

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);
RAm=rank(Am);

(% o84) $$RA = 3$$

(% o85) $$RAm = 3$$

Este sistema sería compatible determinado, ya que $R(A) = R(A') = 3 =$ n.º de incógnitas

7.4. Sistemas dependientes de algún parámetro

Entramos de lleno en el ejercicio típico de 2.º 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:

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

a) Estudia la compatibilidad del siguiente sistema en función del parámetro λ

b) Resuelve si $\lambda = 1$

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

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

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

☞ Creamos las matrices del sistema:

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

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

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

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

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

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

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

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

(detA) $2 - 2\lambda^2$

(% i93) solve(detA=0,λ);

(% o93) $[\lambda = -1, \lambda = 1]$

Si $\lambda \neq \pm 1 \Rightarrow R(A) = R(A') = 3 =$ n.º 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) $\begin{pmatrix} -1 & 0 & 1 \\ -1 & -1 & -1 \\ -2 & -1 & 0 \end{pmatrix}$

(% i95) Am1:at(Am,λ=-1);

(Am1) $\begin{pmatrix} -1 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ -2 & -1 & 0 & 2 \end{pmatrix}$

☞ Calculamos los rangos de $A1$ y $Am1$:

(% i96) rank(A1);

(% o96) 2

(% i97) rank(Am1);

(% o97) 3

$R(A1) = 2 < R(Am1) = 3 \Rightarrow S.I.$

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

(% i98) A2:at(A,λ=1);

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

(% i99) Am2:at(Am,λ=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

$R(A2) = R(Am2) = 2 <$ n.º de incógnitas $\Rightarrow S.C.I$ con $3 - 2 = 1$ 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,λ=1);
e22:at(e2,λ=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 $\boxed{\begin{cases} x = -\mu \\ y = 0 \\ z = \mu \end{cases}}$

☞ 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:

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

(% i105) A3:at(A,λ=3);

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

(% i106) B3:at(B,λ=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. Vectores

8.1. 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

8.2. 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) $$\text{vect}\,(u,v) := \text{express}\,(u \sim v)$$

(% o110) $$\text{modulo}(u) := \sqrt{u.u}$$

(% o111) $$\text{pmixto}\,(u,v,w) := \text{determinant}\,(\text{matrix}\,(u,v,w))$$

8.3. Introducción de vectores

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

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

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

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

8.4. Operaciones básicas con vectores

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

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

(w) $$[-1,1,12]$$

8.5. 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

8.6. Módulo de un vector

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

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

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

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

(% o116) $\mathrm{modulo}(u) := \sqrt{u.u}$

(% i117) modulo(u);

(% o117) $\sqrt{14}$

(% i118) modulo(v);

(% o118) $\sqrt{6}$

8.7. Vectores unitarios

Recuerda: $\boxed{\hat{u} = \dfrac{\vec{u}}{|\vec{u}|}}$

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) $$1$$

8.8. 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) $$\mathrm{vect}\,(u, v) := \mathrm{express}\,(u \sim v)$$

(% 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

8.9. Producto mixto

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

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

☞ 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) $\text{pmixto}(u,v,w) := \text{determinant}(\text{matrix}(u,v,w))$

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

(% o128) 75

9. Configurar arranque de Maxima

En nuestra carpeta personal, hay una carpeta oculta con el nombre .maxima[1], 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);
loga(x,b):=log(x)/log(b);
Mt:5.97e24;
G:6.67e-11;
Rt:6.37e6;
c:3e8;
rad(x):= %pi*x/180;
gra(x):=180*x/ %pi;
load(vect);
vect(u,v):=express(u v);
modulo(u):=sqrt(u.u);
pmixto(u,v,w):=determinant(matrix(u,v,w));
```

[1] Podemos ver en qué carpeta se halla ejecutando maxima_userdir en WxMaxima