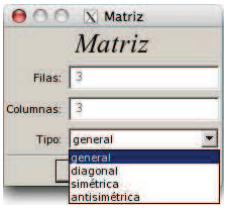
Práctica 1

1.1 Matrices

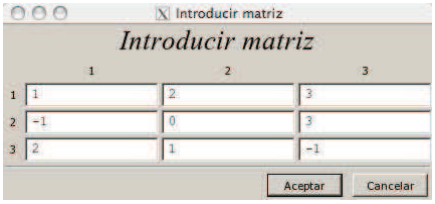
Para introducir una matriz, elemento a elemento, se puede escoger en el menú

Algebra → *Introducir matriz.*

Aparece un cuadro en el que debemos precisar las dimensiones de la matriz, también podemos señalar el tipo de matriz: si es diagonal, simétrica, antisimétrica o el tipo general. Por último, el nombre que vamos a dar a esta matriz.



Al aceptar, se abre un cuadro en el que podemos escribir los elementos de la matriz.



También puede introducirse la matriz anterior con la orden:

```
-->     a: matrix([1,2,3],[-1,0,3],[2,1,-1]);
```

Ejercicio 1. Introducir 3 matrices: las dos primeras, que llamaremos a y b, de dimensiones 3×4 y la tercera, c, cuadrada de orden 4.

La expresión $a[i]$ hace referencia a la fila i–ésima de la matriz a (con formato de lista) y la expresión $a[i, j]$ al elemento $a_{i,j}$. Si lo que se quiere es la fila i–ésima de a con formato de matriz, la orden adecuada es `row(a,i)`. Ejecutar las órdenes:

```
-->    m:a[2]; listp (m); matrixp(m);

-->    n:row(a,2); listp (n); matrixp(n);
```

La sentencia `matrix_size(a)` nos proporciona una lista con las dimensiones de la matriz `a`. Así, `matrix_size(a)[1]` será el número de filas de `a` y `matrix_size(a)[2]` el número de columnas.

1.1.1 Operaciones y funciones con matrices

Suma de matrices $a + b$;
Producto por un escalar $\lambda * a$;
Producto de matrices $a.c$;
Potencia n–ésima de una matriz cuadrada $c\hat{}\hat{}n$;
Determinante de una matriz cuadrada $determinant(c)$;
Rango $rank(a)$;
Matriz identidad de orden n: $ident(n)$;
Matriz traspuesta: $transpose(a)$;
Matriz inversa de c: $invert(c)$;

Ejercicio 2. Probar los comandos anteriores con las matrices introducidas en el ejercicio 1. Observad que si las dimensiones de las matrices que intervienen no son adecuadas *Maxima* nos da un mensaje de error e igualmente nos avisa si la matriz c introducida no es inversible.

Escribir la matriz

```
-->    d: matrix(
        [cos(x)^2,sin(x)^2],
        [cos(x)^2,1-cos(x)^2]
       )$
```

que es obviamente singular. No obstante, la orden

```
-->    invert(d);
```

nos proporciona aparentemente su inversa. Si ejecutamos a continuación la orden

```
-->    trigsimp(%);
```

se detecta el error. Cuando el determinante es 0, *Maxima* nos avisa de que la matriz no es inversible. Sin embargo, si el determinante puede que sea 0 o no, debemos estudiarlo nosotros.

1.1.2 Operaciones elementales por filas

La orden `triangularize(a);` escalona la matriz `a` por filas y la orden `echelon(a);` hace que, además, los elementos iniciales no nulos de cada fila sean un 1.

Ejercicio 3. Probar los comandos anteriores con las matrices

$$a = \begin{pmatrix} 2 & 4 & -1 & -8 \\ 3 & 8 & 1/2 & 0 \\ 2 & 4 & -1 & -8 \end{pmatrix} \text{ y } b = \begin{pmatrix} 2 & 4 & -1 & -8 \\ 3 & 8 & 1/2 & 0 \\ 2 & 4 & -1 & x \end{pmatrix}$$

Utilizar también la orden `rank( )` para responder: ¿Qué podemos deducir de los resultados obtenidos?

Operaciones elementales con las filas de una matriz. *Maxima* tiene dos órdenes para efectuar operaciones con las filas de una matriz: sumar a una fila un múltiplo de otra `rowop(M,i,j,theta)` y cambiar dos filas entre sí `rowswap(M,i,j)`. En ambos casos hace la operación sobre la matriz `M` pero devuelve otra matriz, es decir que no modifica `M`. Nosotros vamos a utilizar aquí las operaciones elementales de modo que la nueva matriz conserve el mismo nombre que la matriz a la que se le hace la operación. Para ello, vamos a utilizar un fichero en el que ya están escritas esas operaciones. Es un fichero con la extensión .mac (**opelem.mac**)

`Archivo: Cargar paquete`

se abre el explorador y se busca. Ahora ya se pueden utilizar esas operaciones sin tener que escribirlas, estas son:

Cambiar las filas i-j en la matriz `a`, $P_{i,j}$

```
cf(a,i,j):=(t:a[i],a[i]:a[j],a[j]:t,a)$
```

Sumar a la fila i-ésima de `a` la j-ésima multiplicada por c, $P_{i,j}(c)$

```
sf(a,i,j,c):=(a[i]:a[i]+c*a[j],a)$
```

Multiplicar la fila i-ésima de `a` por el elemento no nulo c, $P_i(c)$

```
mf(a,i,c):=(a[i]:c*a[i],a)$
```

En los tres casos, la matriz que se obtiene después de efectuar la operación sustituye a `a`.

Ejercicio 4. Introducir una matriz de dimensiones 3×4 para comprobar con ella las tres operaciones elementales por filas.

Ejercicio 5. Utilizando las funciones que acaban de definirse, escalonar por Gauss la matriz

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

1.1.3 Inversa de una matriz por el método de Gauss-Jordan

Recordemos que, si A es una matriz cuadrada de orden n, inversible, al efectuar operaciones elementales en las filas de $(A|I_n)$ hasta obtener $(I_n|B)$, resulta que B es A^{-1}. Consideremos la matriz

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

Comprobamos que es inversible, viendo que su determinante es distinto de 0:

```
-->    is(determinant(a)=0);
```

Para añadirle a continuación la matriz I_3 podemos utilizar la orden `append(a,ident( ))` que añade por filas la matriz identidad. Como nosotros queremos añadir columnas, empleamos también la trasposición:

```
-->    a1:transpose(append(transpose(a),ident(3)));
```

Ejercicio 6. Hallar la matriz inversa de a, por el método de Gauss-Jordan.

Para comprobar el resultado obtenido, vamos a emplear la función
`submatrix`$(i_1, ..., i_m, M, j_1, ..., j_n)$
que devuelve una nueva matriz formada a partir de M pero cuyas filas $i_1, ..., i_m$ y columnas $j_1, ..., j_n$ han sido eliminadas. En nuestro caso, eliminamos las 3 primeras columnas

```
-->    b:submatrix(a1,1,2,3);
```

```
-->    a.b;
```

1.2 Sistemas de ecuaciones lineales

Antes de comenzar con los sistemas borrar los valores anteriormente asignados pues vamos a utilizar para parámetros nombres que antes estaban asociados a matrices.

```
-->    kill(all);
```

Ahora, cargar de nuevo el paquete(**opelem.mac**)

La llamada a la orden `solve` tiene dos listas como argumentos. La primera lista está formada por las ecuaciones a resolver y la segunda por las incógnitas.

```
-->    solve ([x + y = 3],[x,y]);
```

Si la expresión dada no es una ecuación, por no contener el signo de igualdad, se supone que se quiere resolver la ecuación $expr = 0$.

```
-->    solve ([x + y],[x,y]);
```

Esta orden no es específica para ecuaciones lineales. Emplearemos la orden

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

Ejercicio 7. Estudiar los sistemas

$$\left.\begin{array}{rcrcrcrcl} x & + & y & + & z & + & t & = & 1 \\ 2x & + & y & + & 2z & + & t & = & 4 \\ 2x & + & y & + & 2z & + & t & = & 5 \end{array}\right\}$$

$$\left.\begin{array}{rcrcrcrcl} x & + & y & + & z & + & t & = & 1 \\ 2x & + & y & + & 2z & + & t & = & 4 \\ 3x & + & 2y & + & z & & & = & 5 \end{array}\right\}$$

Los sistemas anteriores no tienen parámetros. Vamos a resolver ahora el sistema

$$\left.\begin{array}{rcrcrcrcl} x & + & y & + & z & + & t & = & 1 \\ 2x & + & (a+2)y & + & bz & + & 4t & = & 2 \\ 3x & + & (a+3)y & + & (2b+1)z & + & 8t & = & 5 \\ 4x & + & (a+4)y & + & (2b+2)z & + & 9t & = & 6 \end{array}\right\}$$

en el que a y b son parámetros, no incógnitas.

```
--> eq1:x+y+z+t=1$
    eq2:2*x+(a+2)*y+b*z+4*t=2$
    eq3:3*x+(a+3)*y+(2*b+1)*z+8*t=5$
    eq4:4*x+(a+4)*y+(2*b+2)*z+9*t=6$

--> linsolve([eq1,eq2,eq3,eq4],[x,y,z,t]);
```

Maxima nos indica que la cuarta ecuación es combinación de las anteriores y que, por tanto, la ha eliminado. Se observa además que en la solución no se tiene en cuenta el valor de los parámetros. Así, la solución dada presupone que $a \neq 0$.

Para la resolución sin estas órdenes, procedemos a escalonar la matriz ampliada. Como ya hemos introducido las ecuaciones, en vez de introducir ahora la matriz directamente podemos emplear la orden

```
--> AB:augcoefmatrix([eq1,eq2,eq3,eq4],[x,y,z,t]);
```

que nos proporciona la matriz "ampliada", sin embargo, si observamos el resultado vemos que la columna de términos independientes que añade es la opuesta a la dada, por ello, vamos a multiplicar por -1 la quinta columna

```
--> AB:transpose(mf(transpose(AB),5,-1));
```

Ahora, efectuamos las operaciones elementales:

```
-->    sf(AB,2,1,-2)$  sf(AB,3,1,-3)$  sf(AB,4,1,-4)$

-->    sf(AB,3,2,-1)$  sf(AB,4,2,-1)$

-->    sf(AB,4,3,-1)$
```

Hacemos $a = 0$ que es el caso que tenemos pendiente de estudio

```
-->    AB[2,2]:0$ AB;
```

y seguimos escalonando

```
-->    sf(AB,2,3,-1);

-->    sf(AB,3,2,b/2);
```

Observad que si $a = 0$ y $b = 6$ el sistema es incompatible. Por lo que el caso que queda pendiente de resolver es $a = 0$ y $b \neq 6$.

El problema es el mismo que ya hemos visto cuando se escalonan matrices o cuando se calcula el rango. Así, si las matrices no tienen parámetros podemos emplear las órdenes directas existentes en *Maxima* mientras que, si tienen parámetros, hay que ir controlando los cálculos que se realizan.

Ejercicio 8. Resolver el sistema anterior con $a = 0$ y $b \neq 6$.
Eliminar la fila 4 de AB: `AB:submatrix(4,AB);`
Nueva matriz de términos independientes: `in:col(AB,5);`
Nueva matriz de coeficientes: `ab:submatrix(AB,5);`
Ecuaciones asociadas: `eq:ab.matrix([x],[y],[z],[t]);`
`linsolve([eq[1,1]=in[1,1],eq[2,1]=in[2,1],eq[3,1]=in[3,1]],[x,y,z,t]);`